Thesis Documentation for the Master of Fine Arts Degree

Rochester Institute of Technology
College of Imaging Arts and Sciences
School of Design
Computer Graphics Design

Migration During the Dust Bowl

A thesis project by Devin Kerr

A thesis submitted to the Faculty
of the College of Imaging Arts
and Sciences in candidacy for the
degree of Master of Fine Arts

Thesis Documentation for the Master of Fine Arts Degree

Approvals

Thesis Committee Approval

Assistant Professor Chris Jackson / Thesis Committee Chair

Signature of Committee Chair *Date*

Assistant Professor Jason Arena / Thesis Committee Member

Signature of Committee Member *Date*

Professor Roger Remington / Thesis Commitee Member

Signature of Committee Member *Date*

School of Design Approval

Patti Lachance / Chairperson, School of Design

Signature of School of Design Chairperson *Date*

Rights for Reproduction:

Reproduction Granted:

I, ________, hereby grant/deny permission to Rochester Institute of Technology to reproduce my thesis documentation in whole or part. Any reproduction will not be for commercial use or profit.

Signature of Author *Date*

Inclusion in the RIT Digital Media Library Electronic Thesis and Dissertation (ETD) Archive:

I, ________, additionally grant to Rochester Institute of Technology Digital Media Library the non-exclusive license to archive and provide electronic access to my thesis in whole or in part in all forms of media in perpetuity. I understand that my work, in addition to its bibliographic record and abstract, will be available to the worldwide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles and books) all or part of this thesis. I am aware that Rochester Institute of Technology does not require registration of copyright for ETDs. I hereby certify that, if appropriate, I have obtained and attached written permission statements from owners of each third party copyrighted matter to be included in my thesis. I certify that the version I submit is the same as that approved by my committee.

Signature of Author *Date*

Table of Contents:

Migration During the Dust Bowl

Thesis Proposal:

A two-part exploration into the societal impacts of the Great Depression with supporting teacher documentation and implementation strategies.

Thesis Topic:

My proposed topic will cover the societal issues of the Great Depression. I plan to demonstrate how after the stock market crash, and the downturn in the economy, how everyday Americans lived from day to day. I will go over breadlines, migrations, how people lived, worked and died during the depression era (1929-1939).

Part One:

Part One will be an interactive history lesson based on the time period. The students be able to read the information of the lesson, but also will include embedded interactions that will allow students to apply their acquired knowledge, and will allow teachers to track the progress of their students' learning.

These are sections of content I plan to cover:

Economics & Commerce: This section will outline the reasons for the economic downturn and how Americans dealt with everyday life after the start of the depression. How exactly did Americans loose of their investments and cash?

Proposed Interactions: Students will have a mini-game where they have to create a weekly budget and see how far they can stretch a dollar.

Everyday Life: Will deal with the everyday struggles families and individuals faced during the Great Depression. Breadlines, working odd-jobs for spare change, and the sacrifices they had to make in order to survive from day to day.

Proposed Interaction: A side by side comparison chart of how Americans live today as opposed to the 1930's in terms of nutrition, income, living conditions and other life essentials. Then students will be challenged to formulate their own conclusions.

Migrants & Migration: Students will learn the reasons for large communities of people uprooting from their existing lands, and moving to new areas or states to find new opportunities. Will discuss some of the more popular trends of the time, like hitch-hiking, the "Migrant Trail" in California and

Proposed Interaction: A Map of the United States that students can view a year by year progression and view the population movement, and the trend of Americans leaving the rural areas of the Midwest towards the direction of industrial areas and California.

Government: How exactly did the government help Americans get back on their feet? Through the blinding neglect of the Hoover administration to the various programs instituted by Roosevelt and his “New Deal”, what worked best and instilled the American spirit back to the downtrodden.

Proposed Interaction: Students will have a drag and drop challenge where they have to match the governmental programs with the causes and affects of these programs or policies. Each correct answer will score points for the students, whereas each wrong choice will score negative points.

Part Two

Part Two will be an interactive role playing game based loosely on the game “Oregon Trail” the book “Grapes of Wrath” and the content covered in Part One of my thesis. Students will play a family trying to navigate their way across country from Oklahoma to California for the hopes of a better life. Initially presented with a budget, they will need to plan how to make it across country and for what they might need, as well as saving some in the instance they might need to buy items along the way.

They will be faced with some critical situations that they must deal with, from their automobile breaking down and needing repairs to having members of the party dying along the way.

Game Strategies: Students will travel a set distance before their budget runs out. Everyday on the road will cost them a dollar for gas, as well as any food they might need to make their journey. Students will plan for the unexpected situations, that will be a time/cost penalty.

Additionally, students will have the opportunity to make money by doing odd-jobs, forage for extra food, and to interact with people along the way to get to their ultimate goal. This however will cost them a penalty of a day/dollar and sometimes they might not be able to recoup these losses.

Game Features: Students will be travelling along Route 66, and will be able to track their progress along the route. There will be a menu to access their budget information, food amount, members of the party, and additional items that might be brought along.

Part Three:

Part Three is going to be based purely on the technical aspects of Flash, providing a tracking tool for teachers to measure student progress. Because I am planning to track student data via a database on how well they perform the interactions within the first two parts, I hope to make an interface where teachers can see where individual students might need more assistance in grasping the concepts and goals.

I will also include the lesson plan I develop for the content as a guide for teachers on how to develop their own curriculum for their classroom based upon the material presented. This could help teachers that have not instituted multimedia materials previously, or on how they could incorporate additional materials or ideas into the lesson.

What I plan to track:

- Student Scores on each interaction within the lesson plan, as well as on the game.
  This will allow teachers to see how well students are grasping materials and concepts.
- Time spent on each section, just so teachers can see that students are reviewing the materials and spending time.
- Student questions on the material. This lesson could be assigned as homework, and students could post questions that the teacher could answer during a discussion period.

Graphic Style:

The design style that I'm choosing to produce my thesis in is going to be in a illustrative style derived from photography of the era. This is due in part because many of the images from this time period are the insightful photographs produced by the photo-journalists of the time. Also the work of Lester Beall and other prominent designers of the time will influence the graphic style.

The colors of the interface will be muted and earthy. Also there will be the photographs, which will be either grey and sepia toned to preserve their historical context and atmosphere.

For the game, I am envisioning a style similar to the Oregon Trail game, where students will have a foregorund/background view of the world as they travel from Oklahoma to California. I am choosing to work with the cleaner look of vector art for this section.

Project Rationale:

The field of education is rapidly embracing technology and the internet as a tool of learning, this product would fit nicely with the blended learning environments these school districts are pursuing. Instead of having students learning through purely a text-book based curriculum, by embracing the students ability to learn by using their auditory and visual senses, as well as literacy, they'll be able to absorb more content, and enjoy the content at the same time.

Target Audience:

My target audience is junior high school students ages 11-13. Because this topic is generally covered in History class, this could serve as a tool to help aid in building their knowledge of the Great Depression. Secondarily, the game could be marketed towards children ages 8-17 with a passion towards history.

Migration During the Dust Bowl

Project Considerations: “Migration During the Dust Bowl” was inspired by many sources during my initial concept stages. The desire to illustrate a crucial period of American history, a growing interest in interactive media, a need to tell the story of difficult times.

The steps towards my thesis development were arduous, plodding. The concept for the game came from the first forays into outlining my thesis. Initially, I had proposed a series of quizzes and interactions that were to play off some of my experiments on other projects, including the Battle of Saratoga, and Migration Patterns In the Great Depression. Some of these interactions included maps of battle terrains, XML data sets and 3-D models of soldiers and equipment. Though this was a start towards the final development of my thesis, it needed further revision and became what it is today.

Historical Considerations: John Steinbeck, in his book “The Grapes of Wrath” encapsulated the essence of what I was going for in the development of my thesis. Though his book is fiction, it has elements of historical trends and moods of the era. Being a writer in California during the time, his first hand observations of migrant farmers, and the conditions they endured greatly influenced the tone of his book. It was from this inspiration from which I took the overall tone for the game.

Design Considerations: The overall design scheme for the game is photo-realism, based off information gathered from the time. Influenced by the descriptions of migrant workers and period conditions in the *Grapes of Wrath, The Plow that Broke The Plains* **and other source material, the choice for photo realism was self evident. Portraying the conditions in this style allows for the inclusion of photos, while not relying heavily on their use to convey information**. My target audience is junior high students and I wanted them to have the feel of the dust and dinginess of the time, without getting too deep so as to distract them.

In designing the components to the game, there were many considerations to take into account. Most of my inspiration for the imagery of the game came from photos of the people and places of the time. I was heavily influenced by photographs taken by Dorothea Lange and Walker Evans, two noted photographers of the time. An image that could be considered to encapsulate the period best is Dorothea Lange’s “Migrant Mother”. Their images were people-centric, concerned about portraying the issues of the time. My examination of these images influenced the design of the game characters, through simple dress and coloration.

Although most of the photos of the time were black and white, I chose to design this game in color. In my experiences, my target audience wants education in an engaging fashion. By building the game in color, this makes the family of characters more relatable and appealing. It also breaks the stereotypical portrayal of the times. The majority of media of this time period is black and white. Using color allowed me to give the game a vibrance that it might otherwise not have.

Educational Content:

As stated before, Steinbeck and his work was main inspiration for this game. The book though fictionally based, was very educational. To elaborate on the facts I had to turn to junior high textbooks. By using recommended state textbooks, I was able to keep a continuity of message between the game and what the students were learning in class.

In addition to the textbooks, I compiled my information from a variety of sources. Public Broadcasting Services' ***The American Experience*** series produced programs based on historical information on the Dust Bowl, and had an informative website with lesson plans for educators to implement into their own classrooms. Using these resources allowed for the easy breakdown of material for use in a classroom setting.

This time period also has held such great interest in my own studies, that some of the content was initially conceived through sources I have come across in my own personal time.

Migration During the Dust Bowl

Game Idea: The idea of the game is to have students complete a journey from Oklahoma City, Oklahoma to San Bernardino California during the year 1937. As migrant farmers, they are choosing to make this journey for the promise of a better future for themselves and their family. Along the way, the students will need to experience some of the hardships these families faced.

Game Play: The target game running length is 15-30 minutes. This goal has been set to allow students to play during a normal class period.

Introduction: Upon opening the game, the student will be presented with a brief history describing the Dust Bowl and the lives of migrant farmers during the 1930's.

After this initial screen, the students will be presented with their challenge. As migrant farmers living near Oklahoma City, Oklahoma, years of bad cotton crops have forced the bank to foreclose on their farmland. They have just a few days to leave the farmhouse before they are evicted by the bank, and their house is plowed into the soil, and the land becomes a part of a larger corporate industrial farm run by an agriculture conglomerate.

Rumors have been overheard by your family that there are an ample amount of jobs out in California picking fruit and other crops, that you decide to make the journey out there in hopes of a better future for your family. Before you makes this journey however, you need to prepare what you can, and sell the rest off to have enough money to make the journey out west.

Starting the Game: After the introduction, the student will have to enter their name into the opening screen. This student will then become the "Head of the Household" for the remainder of the trip. Everything is reliant on this one person reaching their goal of making it to California.

Once the student enters their name into the screen, they will have the option of picking their family and role within this family. They will have two options to choose from.

- A father, mother, 2 sons and a daughter
- A father, mother, 2 daughters and a son

The students can then customize their party by choosing names for the members. These names will all be tracked within the database.

Preparing for the Journey: Initially, students will need to decide what they will need to bring with them on their journey. They will have to choose from bringing pots, pans, clothing, house essentials... etc

They will start off with $50 to budget towards gas and food for their journey
Gas – Gas will stay within a range of throughout the trip.
Food – potatoes, canned fruits & vegetables, salt pork, flour, sugar, coffee

With food, students will have to provision each item in pounds. In addition to food, students will have to choose from a few items from the homestead they are leaving to bring along their journey.

What students can choose from:

2 pots – one small, one large
2 pans – one small, one large
2 jugs for water – one 5 gallon, one 10 gallon
Changes of clothes – either one day, two days or three days worth
Cleaning supplies – soap, rags
Eating supplies – dishes, spoons, forks, knives, cups
Sleeping supplies – blankets, pillows
Personal Items – books, toys, farm equipment, mementos

There will be limited space in the car, and food and other items will have specific weights. They will not be able to go beyond the car's weight limit.

Item Weights and the Weight Limit

The weight limit will be 100 pounds maximum of items they can bring with them. There is two reasons for this; to prevent students from bringing too many items along with them, as means of selling items along the way and; the more weight in the car results in decreased fuel mileage.

Specific Weights:

Pots:	small	1 pound
	large	2 pounds
Pans:	small	1 pound
	large	2 pounds
Water:	5 gallons	40 pounds
	10 gallons	80 pounds
Clothes:	one day	1 pound
	two days	2 pounds
	three days	3 pounds
Cleaning supplies:		1 pound
Utensils:	dishes & cups	1 pound per setting
	spoons forks & knives	1 pound per setting
Sleeping supplies:	5 blankets& pillows	1 pound

An ideal weight set-up might be as follows:

5 Gallons of Water	40 pounds
Small Pan	1 pound
Small Pot	1 pound
One day of clothes	1 pound
Cleaning supplies	1 pound
Utensils	1 pound
Sleeping supplies	1 pound
3 days worth of food	15 pounds
Total:	61 pounds

The amount of weight a student is traveling with will be tracked, and will result in the amount of mileage a student gets on each tank of gas.

Amount of Travel:

A rough estimate of the amount of travel the students will have to travel is 1300 miles. A day will elapse after either 200 miles has been reached, the student enters a city along the route, or if a penalty has been incurred. In a perfect situation, the journey will last 11 days, 7 for travel, and 4 cities in between the route.

Consumption of Food:

Each member of the party will consume 1 pound of food a day (5 members = 5 pounds, 3 members = 3 pounds... etc). In the event that the party runs out of food, they will have only one day to reach a town, by either back tracking or pushing forward. If they do not reach a town within a day, the “Unforeseen Circumstances” counter will start, and will randomly pick off members of the party.

Consumption of Gas:

While traveling, there will be a series of counters running to monitor progress. One will be miles traveled, which will be displayed upon the top, incrementally in units of 10. The automobile will be able to travel from a range of 100-150 miles between each 10 gallon fill-up. The amount of weight the student is travelling with results in the maximization of gas mileage.

50 pounds of weight and under =	15 miles per gallon
51–60 pounds of weight =	14 miles per gallon
61–70 pounds of weight =	13 miles per gallon
71–80 pounds of weight =	12 miles per gallon
81–90 pounds of weight =	11 miles per gallon
91–100 pounds of weight =	10 miles per gallon

There will be gas stations placed from 25-75 miles apart, and the price of gas will fluctuate from the mean amount. It is up to the student to purchase gas at these gas stations.

- If the student purchases gas: The gas tank counter will be reset back to zero.
- If the student does not purchase gas: The gas tank counter will continue to run.

If the student runs out of gas, this will cause a day to elapse in the time line, which will cause the consumption of food and water.

Breakdown of Budget Costs:

Students will start off with 50 dollars to make their journey. From this, students will need to save enough money to make the journey safely across the country without incident.

- Gas: the mean of the cost of gas in 1937 is 15 cents per gallon, though will vary from 14 cents to 17 cents at different gas stations.
- Flour: Will cost 20 cents (mean cost) per pound, though will vary from 18-23 cents per pound
- Potatoes: Will cost 25 cents (mean cost) per pound, though will vary from 23-28 cents per pound
- Vegetables: Will cost 30 cents (mean cost) per pound, though will vary from 28-33 cents per pound
- Pork: Will cost 40 cents (mean cost) per pound, though will vary from 38-43 cents per pound
- Coffee: Will cost 35 cents (mean cost) per pound, though will vary from 33-38 cents per pound
- Sugar: Will cost 45 cents (mean cost) per pound, though will vary from 43-48 cents per pound

Reasoning for Food:

Students will have to have to purchase a full compliment of food-stuffs to complete the journey. A lack of any of the main food stuffs (Flour, Potatoes, Vegetables, Pork) will the "Unforeseen Circumstances" counter. The reasoning towards this is to prevent students from just making the trip with just flour as a food source. Coffee and sugar are included to deceive students towards wasting their budget resources for non-essential items.

Total Costs of Journey:

An ideal budget might consist of this (given at mean costs)

Gas: 130 gallons (10 m.p.g.) at 15 cents =	$19.50
Flour: 14 pounds at 20 cents =	$2.80
Potatoes: 13 pounds at 25 cents =	$3.25
Vegetables: 13 pounds at 30 cents =	$3.90
Pork: 13 pounds at 40 cents =	$5.20
Total	$34.65

This total will vary though, as prices will vary from location to location and other budget issues that may arise. On average, students might expect to spend between 35–40 dollars towards the total cost of the journey.

Unforeseen Circumstances: The "Unforeseen Circumstances" counter will be used to create challenges to the student. This will run on a random number count in a still yet to be determined range. If the player encounters the number one in the random number selection, then this will call a function to randomly choose between a list of selected challenges. Each of these challenges will be weighted so it becomes more likely to have a flat tire then to have a member of the party die on route. Each penalty incurred will have either a monetary or time penalty. This includes:

- Flat Tires= $2.50 cost
- Overheated Radiator = $5 cost, 1 day penalty incurred
- Blown Head Gasket = $10 cost, 1 day penalty incurred
- Sickness in Party = 1 day penalty incurred
- Death in Party = 1 day penalty incurred

Deaths in Party: The challenge of the game is to get as many family members across the country, and thus deaths in the party are a required part of the game. There are only two ways to have family members die in route, either due to poor planning, or unexpected circumstances.

If the party runs out of food, a counter will start that will start to kill off family members, until food is bought.

When the "death in party" penalty is incurred, once the number reaches 5, the game will end.

Towns Along the Road: Students will have to stop in 4 major towns along their way to San Bernardino California. These will be places to replenish supplies of food, water and to gather information for the journey ahead.

Upon entering the town, the students will be presented with a brief history of the town they are entering, along with time-period photographs to give historical perspective to each of the towns.

Amarillo, Texas: 300 miles into journey
Pie Town, New Mexico: 600 miles into journey
Flagstaff, Arizona: 900 miles into journey
Needles, California: 1100 miles into journey
San Bernardino California: 1300 miles into journey.

As they travel on the road, there will be certain historical "markers" that the students can choose to read about. These will assist in giving the student more information on the journey they are undertaking, or will increase awareness of other relative historical information along the journey. For example, before Pie Town, students will get info on the Civilian Conservation Corps, one of President Roosevelt's New Deal programs, aimed to help decrease the American unemployment roles and the lift the spirits of the people up.

When the students reaches a town, there will be a brief description of the town, as well as a little historical note about the stretch of travel they just completed. Each town contains an Interactive Challenge lesson, and a key element that the student must collect to complete their journey. While in town, students have the option of visiting places to replenish supplies, get gas and other information they will need along their journey.

Historical Markers:

Historical Markers will be used to provide insight to the different areas the students will be traveling through. These will be placed in between cities along the route to give historical, social and geographical perspective to where the students are travelling. These markers will be optional along the route, though not reviewing the content will cost the student points towards their overall score.

- **Route 66:** In between Oklahoma City and Amarillo
- **Civilian Conservation Corps:** In Between Amarillo and Pie Town
- **The Works Public Administration :** In Between Pie Town and Flagstaff
- **The NRA and AAA Acts:** In Between Flagstaff and Needles
- **Migrant Hardships:** In Between Needles and San Bernardino

Interactive Challenges:

Interactive Challenges will be used to help reinforce the content and the message of the Dust Bowl, Historical figures and events and the programs aimed to help lift the American economy and spirit. These Interactive Challenges will be implemented in each town and are non-optional. Students will learn about a variety of topics that will help increase knowledge and awareness of the time period, as well as develop an understanding of the topics through interactions such as drag and drop, item sequencing and multiple choice questions in a unique format

The interactions that will be in each town are:

- **Amarillo, Texas:** Soil Strata demonstration, and how the effects of agriculture, irrigation and poor land use created the Dust Bowl.
- **Pie Town, New Mexico:** A play on the “Whack A Mole” concept, where students will have the opportunity to “whack” the correct historical figure.
- **Flagstaff, Arizona:** Hoover Dam sequence puzzle where students will have to properly sequence the building of the hoover dam correctly to produce the correct outcome.
- **Needles, California:** Exploration of famous chroniclers of the time, where students will have an opportunity to learn about famous photographers and songwriters and how they interpreted the conditions of the time.

The town of Amarillo, Texas: 4 buildings to explore

Grocery Store: to re-supply with food and other needed items.
Gas Station/Car Repair: Fill the car up with gas, and for automotive repair
Soda Fountain: Talk to a local or two as you whet your whistle
Five and Dime: If you brought some things from the homestead, you'll be able to sell some items off for some extra cash

The key element the student will need to collect in Amarillo is a photograph and a letter from a man in the saloon. His son went out west to San Bernardino, and would like you to deliver this message to him.

The town of Pie Town, New Mexico 4 buildings to explore

Grocery Store: to re-supply with food and other needed items.
Gas Station/Car Repair: Fill the car up with gas, and for automotive repair
Local Community Center: get a free sandwich and find out some information from other travelers.
Five and Dime: If you brought some things from the homestead, you'll be able to sell some items off for some extra cash

The key element the student will have to collect in Pie Town is a map of the farms in Southern California. This will be located in the Farm Supply Store. The map is not necessary for successful game completion.

The town of Flagstaff, Arizona: 4 buildings to explore

Grocery Store: to re-supply with food and other needed items.
Gas Station/Car Repair: Fill the car up with gas, and for automotive repair
Five and Dime: If you brought some things from the homestead, you'll be able to sell some items off for some extra cash
The Soup Kitchen: get a free meal for the family.

The key element the student will need to collect is a handbill for picking oranges in California. This will be located at the soup kitchen. Also they will need to remember the name of the old man who gives them the handbill and the secret code word, for this is needed to work on the orange farm.

The town of Needles, California: 4 buildings to explore

Grocery Store: to re-supply with food and other needed items.
Gas Station/Car Repair: Fill the car up with gas, and for automotive repair
Local Community Center: get a free sandwich and find out some information from other travelers.
Five and Dime: If you brought some things from the homestead, you'll be able to sell some items off for some extra cash.

The key element the student will need to collect in Needles is a registration form at the Community Center to become a resident in California. This will be needed to be dropped off in San Bernardino to collect government assistance and to get a workers card.

The town of San Bernardino, California

4 buildings to explore

Grocery Store: to re-supply with food and other needed items.
Gas Station/Car Repair: Fill the car up with gas, and for automotive repair
Local Community Center: get a free sandwich and find out some information from other travelers.
Five and Dime: If you brought some things from the homestead, you'll be able to sell some items off for some extra cash.

Once your group makes it to San Bernardino, you'll go to the local community center to drop off your residency form, get your workers card and to inquire about the jobs that have been advertised back in Oklahoma. The storekeeper will refer you to go a few miles out to town, and look for the collection of tents.

When you arrive in the tent city, you'll be faced with the grim depiction of life as a Californian migrant farmer. You'll be able to talk to a few people as they walk through the tent city.

Game Debriefing

Upon arriving at San Bernardino, this completes the travel portion of the trip. At this juncture, students will be presented with a 10 question quiz as a way to reinforce the content they have learned through the journey. Questions will be presented in a multiple choice format and the overall score will reflect in the calculation of the final score.

Game Completion

There are two outcomes to the game once you reach San Bernardino.

- If you have collected all the clues along the journey

You will have to locate the man for who the letter you received in Amarillo is for. When you locate this man in the tent city, he will explain the way to get to the orange grove. Upon arriving at the orange grove, you will have to show the handbill and give the code word to the farmer.

If the code word is right, and you have the workers card, he will hire your family on as hired hands, and will give you a bungalow for your family to live in. This will ensure a stable place to live, as you will be a tenant farmer living on the land.

- If you do not collect all the needed clues

Your family will live in abject poverty as it tries to search for the limited jobs available in California. Part of the requirement to receive government assistance is to have a residence, so without the letter, the handbill, the code word and the residency form, you will be unable to receive assistance.

Migration During the Dust Bowl

Game Play:

The first part of the proposed thesis is to create a game based on the time of the Great Depression. To complete this task, I have chosen to create the game using a Macromedia Flash framework, in conjunction with other programs to create a cross platform game that all students will be able to play.

The technology apart from Flash that I have chosen to use in game creation are:

- XML: for use in graphic placement, use in storing an retrieving information. It will allow for the game information to be changed without having to alter the original Flash file.

With the proposed use of technology, I plan to build a robust and dynamic game that will gather information on the participating students, and will create a game log to be assessed by the teacher assigning the lesson.

During Game Play:

Starting the game off, when a student enters his or her name into the input field, a new entry will be entered into the database. From this, it will track how well the students' budget their money, in how much food and supplies were bought.

Along the journey the students progress will be tracked, and information sent to the database. For each town the students enter, a checklist of what buildings they enter, and if they gathered the clues necessary for successful game completion will be tracked also. Each historical marker and interactive challenge viewed will be counted towards the final calculation.

At the end of the game, information will be gathered on how well the student did during their journey. It will tabulate information on the party members, how many days it took, how much money/food they have left and the job they received.
If a student does not complete the journey before all the family members die, information will be gathered about how far they went, and what information was received by the end of their game.

Delivery Method:

CD-ROM delivered, downloaded onto computer for seamless play. Access to online version so students can participate at home.

Technology Targets:

PC: 1 ghz Pentium 4 processor with 512 megabytes of RAM

MAC: 1 ghz G5 processor with 512 megabytes of RAM

Display: 1024x768 display, thousands of colors. 8 megabytes of video RAM

Flash Player: Flash 8 player needed for display of graphics.

Internet: Game play requires connection to the Internet. T1/DSL/Cable modem technology is recommended for seamless uploading of information.

Applications Used:

The game shall be run through a Macromedia Flash Interface. This will allow content to be loaded dynamically, and will allow for smooth operation of the game and interface. Flash technology will also allow for the implementation of XML into the game structure. Graphics used in the game shall be generated in Adobe Photoshop and Illustrator.

Flash Actionscripting:

Actionscript requirements will be as follows:

- Use of XML parsing scripts to place information and graphics in precise locations.
- Use of randomization scripts to generate game play variables as described above
- Use of input text fields for students name, information and student customization of the game.

Flash and XML Implementation:

The goal is to make this application driven solely through the use of XML. Variables entered through XML forms generated by hand into Dreamweaver will set up the graphical interface, and will set game play rules and challenges.

Proposed Information Collection Markers

During game play, students will be tracked on their progress regarding:

- Family members
- How far they've traveled on route to California
- What Historical Markers have they explored
- How many interactive challenges were completed
- How many clues were collected
- How well they have budgeted their allocated money
- Their end results:
    - Final debriefing quiz score
    - Remaining balance of cash
    - Their employment situation once they have in California

Migration During the Dust Bowl

Target Audience:

Students in 7th and 8th grade studying the effects of the Great Depression upon American Society.

Prerequisites:

Students must have a rudimentary understanding of American History leading up to the Great Depression.

Students must have a preliminary lesson and discussion of the topics causing the Great Depression.

Rationale for Lesson:

Books and videos can only evoke responses from students on base levels. Through the application of this game, it seeks to reinforce the materials presented through previous lessons. The goal of this lesson is to emotionally involve the students in the learning process through a medium that they are comfortable with.

Teaching Strategies:

Role Playing:
Situation Gaming:

Evaluation Process:

Students will develop skills on how to budget for food and gas along the journey towards creating skills for budgeting finances.

Markers along the journey will track the students progress, and will be collected for review by the teacher tracking tool.

Students will be graded on their ability to grasp the material, and to answer questions based upon material posed by the teacher.

Source Material:

The Grapes of Wrath, by John Steinbeck.
Woody Guthrie’s ballads depicting the lives of migrants and the time period.
School district supplied textbooks.
Videos on the issues and history of the time period.
Writings of migrant farmers who made the journey.
Works of Photojournalists of the time.

Materials / Technology Needed:

PC: 1 ghz Pentium 4 processor with 512 megabytes of RAM

MAC: 1 ghz G5 processor with 512 megabytes of RAM

Display: 1024x768 display, thousands of colors. 8 megabytes of video RAM

Flash Player: Flash 8 player needed for display of graphics.

Internet: Game play requires connection to the Internet. T1/DSL/Cable modem technology is recommended for seamless uploading of information.

Migration During the Dust Bowl

Goals: At the end of this interactive section, students will know:	**Objectives:** At the end of this interactive section, students will :	**Strategies:** How students will learn about the goals and objectives:
Will learn the hardships faced by migrant farmers during the Great Depression	Through situation gaming, students will grasp lessons on planning a budget and allocating resources for food and supplies. Through role playing, students will become a migrant farmer trying to make the journey from Oklahoma.	Through peer interaction, students will be able to understand how fellow peer successes / shortcomings influenced the completion of the required tasks. Historical information encountered through the journey will round the students understanding of the time period. Depictions of hardships faced by migrant families will influence the students feelings of the time period.
Will learn that through interactions with community members can aid in the journey, and can garner valuable information.	Through decisions made by students, understanding that the migration was not easy to complete will be generated.	Students will utilize networking skills with the communities they visit which will allow them to complete the game.

User Testing Phase One:

After the completion of a fully working prototype in early May of 2007, user testing was the next step in the development process. The early testing process involved individuals more versed with interactive media, who could isolate problems with the interface, content issues and overall game play.

Early issues that arose through this testing were the following

- Game play was too complex for the target audience, with too many variables to keep track of for overall game play.
- Issues with game length, and in particular the time it took to travel town to town.
- Overall content needed to be simplified for target audience.

In addressing these issues, I looked into what parts I could strip out of the code, that would not affect overall game play, but would enhance the experience. I identified some areas that I addressed in the next prototype

- Removing the necessity to purchase supplies at the Five and Dime store, and made these optional, The user would start off with a randomized amount of supplies that they could then sell for extra cash. This would make it more of a buy and sell option to make more cash on the side for the trip.
- Speeding up the road portion of the game. I increased the amount of the variable that would track distance accumulated in the function from .25 miles per calling of the function to .5 miles per, thus doubling the rate.
- Addressed the content issue by breaking up sections into more simplified explanations, and included more imagery.

User Testing Phase Two:

With the initial round of revisions done, I had approached a former professor at another school about exhibiting my project for her class. Feedback from this session included the following.

- The need for the instructions to appear at each stage they are needed. It was initially difficult to understand the need for food, or that they needed to budget money for the road ahead.
- Simplification and better instructions for the interactions.
- Better describe the need for talking with people, and using call outs to initiate the user to converse.

This round of user feedback was good blind test of the program, as though I had introduced the project and plot of the game, I had allowed them to play the game on their own time, without my instruction. This enabled them to find problems addressing the game play that were often overlooked by myself during the design process.

Ways I addressed these issues were to:

- Simplify the Hoover Dam interaction to include clues as to the steps one must take to complete the puzzle. Before this addition, the 8 choices did not have descriptions, so people would oftentimes not understand the sequence.
- At the first scenes of the grocery store, the five and dime, Route 66 and entering towns, the instructions would open to the relevant section. This would give users clues on how to proceed with these screens.
- Characters present in certain sections have a mouse over prompt to click on them to open a dialog window.

User Testing Final Phase:

The last round of user testing took place after my thesis project was exhibited, having students and teachers from 7th and 8th grade history classes at Great Hollow Junior High School in Smithtown, New York. The quality of the feedback received during this session, though good, turned out to be very limited to the scope of just what the content covered. Though the Dust Bowl is an important part of the history of the Great Depression, the content and message did not connect with many of the students, whether from the student having a bias to history in general, or that the story told was not applicable to the history of what families faced in the New York metro region of the time. The feedback I received from the teachers I found to be more positive on the message of the game, and the fact that it is an essential part of the Great Depression that does tend to be reflected less in the textbooks, as the New Deal programs tend to dominate the content written about the time.

Feedback thus from these sessions were from two perspectives. Student feedback included:

- Speed up the game from the aspect of travel.
- Break down content into bullet points.

Teacher feedback included:

- The game is fun to play, though would not fit well in a non-wired classroom. The game would need to be implemented through a computer lab session, or assigned as homework.
- Target areas in the Midwest where this part of history is covered more in detail, and would be more relevant to the students.

Though I am actively looking to improve my project at every step, I have run into some issues implementing some of the feedback received from a technical end. I haven’t been able to figure out how to place line breaks, and thus bullet points into an XML document without having to restructure sections of the project. I have taken these into consideration, and would like to make note that if the opportunity did present itself, these changes would be made.

Audience Feedback:

Audience feedback from my thesis project is hard to encapsulate into just a single point. There are those who enjoy an interactive challenge, those that can be intimidated by technology, those who love anything historical, and those for which history is a non starter. My challenge was to try to bridge the voids that can separate these groups and to create an enjoyable experience for the user.

Feedback I had received from my testing and thesis exhibition can be summarized into two groups. There were those who took to the content and messaging of the project in positive way, and were more than happy to leave feedback and support. There was also the group that did not engage with the project, whether from the point of things took too long to develop, or they had little to no interest in the subject.

Positive feedback from users included:

- Fun and challenging game play
- The graphic qualities were appealing to the audience.
- The interactions, especially the Whack-A-Mole section was fun to play

Negative feedback and observations of users included:

- Game took too long to develop in terms of end result. Too many interruptions, from interactions, to towns and travel issues disrupted some users.
- Too many times users had to purchase gas, and would frustrate them.
- Did not understand the need to talk to people to affect the outcome of the game. Users after initial interactions with store owners, would not question them in subsequent stores, thus missing opportunities to better the outcome.

A surprising fact I had observed during my exhibition that the project attracted groups of users that I had not initially foreseen engaging in a project of this type. Users, typically ages above 50 were actively engaged with the project, and were some of the people who played the game to completion. Additionally, other adult users were the ones to ask the most questions, and actively seek to complete sections of the game.

Conclusion:

Throughout the scope of the project, I strove to build a robust, comphrehensive game, based on sound principles of interactive design and educational pedagogy. The goal was to build a education game, engaging to the user in a heuristic manner, and enabling them to drawn their own conclusions of the time period based on the content provided. I feel in these areas I have achieved the goals I had set forth.

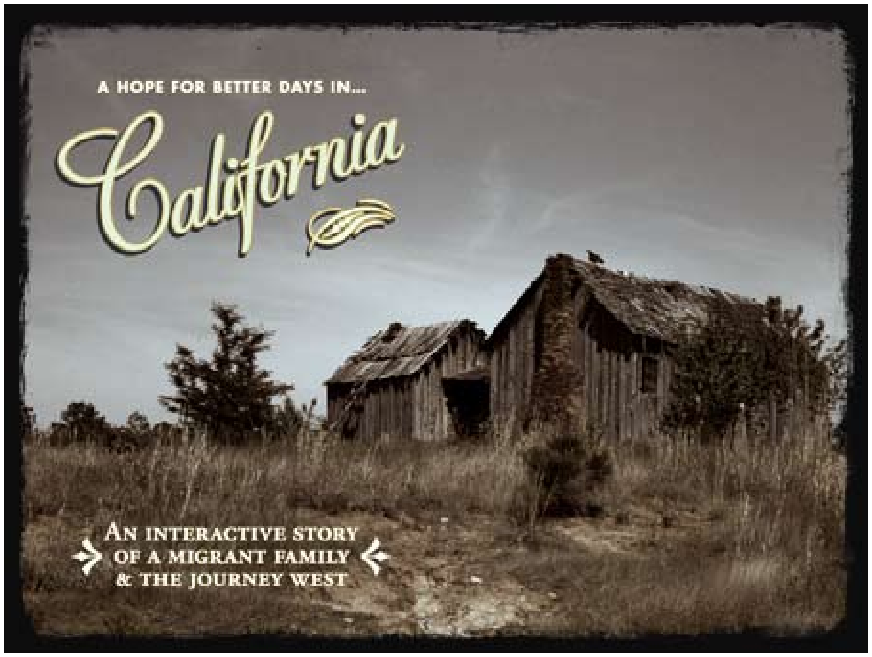
The significance of this project I feel will present itself in the years to come. As more and more educational institutions strive to place content and classes online, being able to deliver engaging user experiences will allow for them to differentiate themselves form the competetion. Though this thesis project was an exploration into such a concept, there are practical applications of the project that could be used for such instances. With further development, the addition of being able to track measureable results would be much desired for the practical application of such a project.

The development process of the “Migration During the Dust Bowl” thesis project has been more that just a learning experience for myself. It has caused reflection into the steps and methods I take towards the development of a project, and has allowed for growth in areas of graphic and interactive design that I had once not considered strengths, and had opened new opportunities. Possibilities of expanding my explorations into Illustrator based vector components into Flash interfaces, expanding the depth and knowledge of my Actionscript abilities, or other avenues that have opened throughout the process.

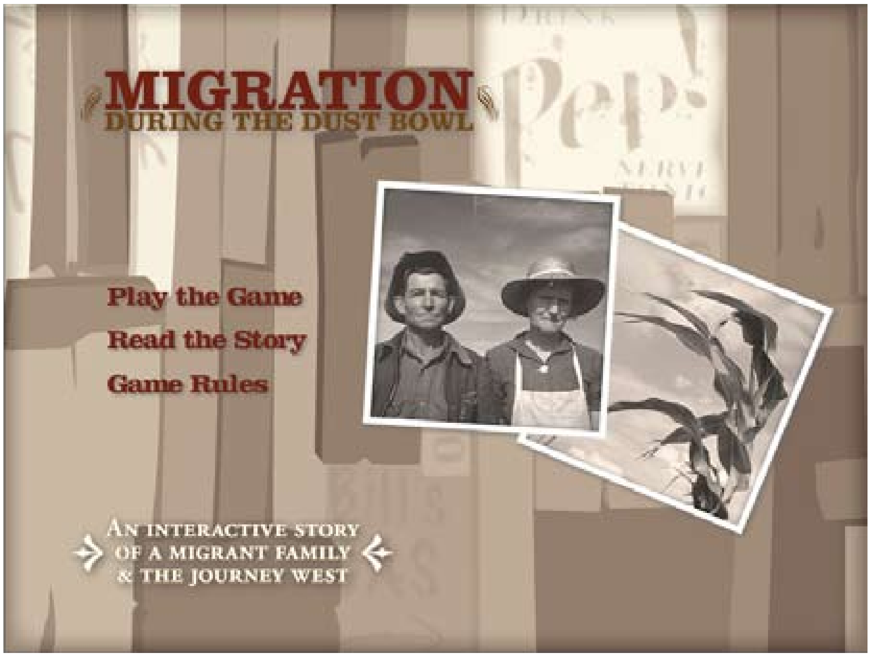
This said, there have been new questions raised during the process, and since the completetion of the thesis. How can I make my game more engaging? How can I streamline my process further, as to shorten the development of a practical prototype? All important questions, that with the proper time and reflection into the methodology undertaken to accomplish my thesis project can only answer.

Graphic Development Process:

Initial Concept for Intro screen, based off period photography



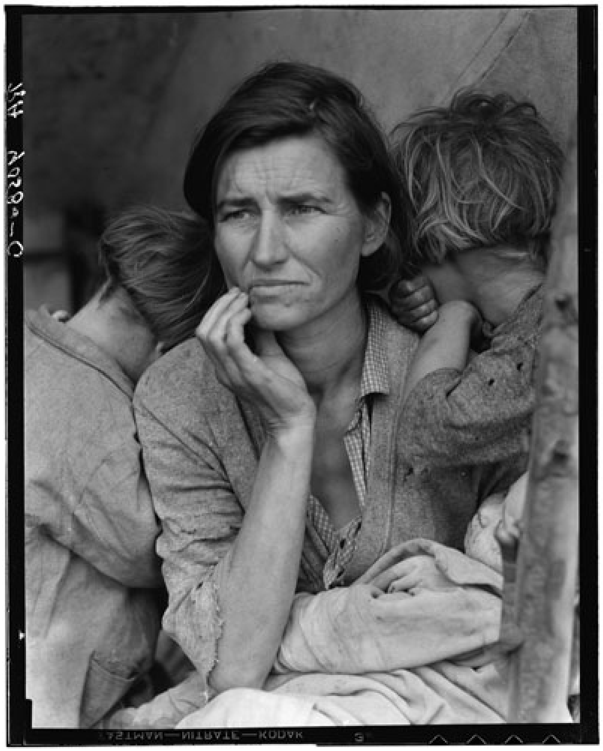
Final Screen concept using vector style



Migration During the Dust Bowl

Graphic Development Process:

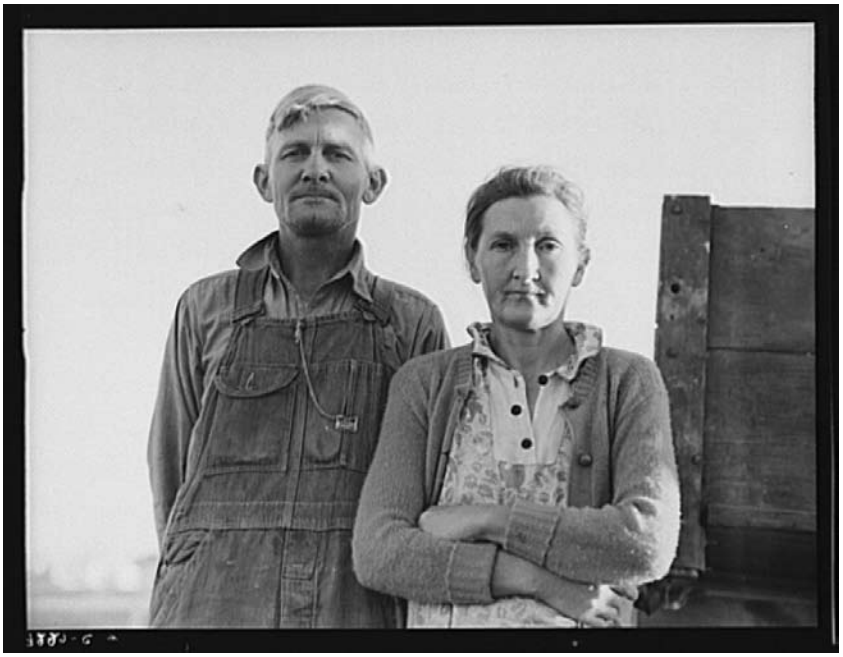
Photography that served as inspiration for the graphics



Migration During the Dust Bowl

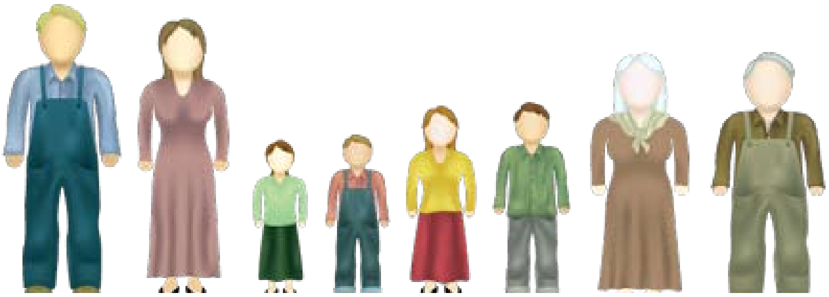
Graphic Development Process:

Additional photography that served as inspiration for the graphics

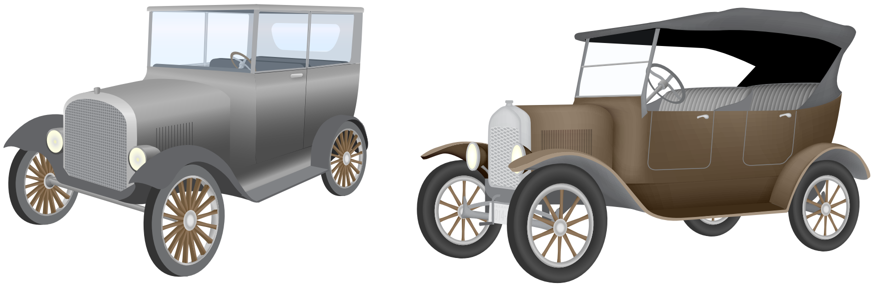


Graphic Development Process:

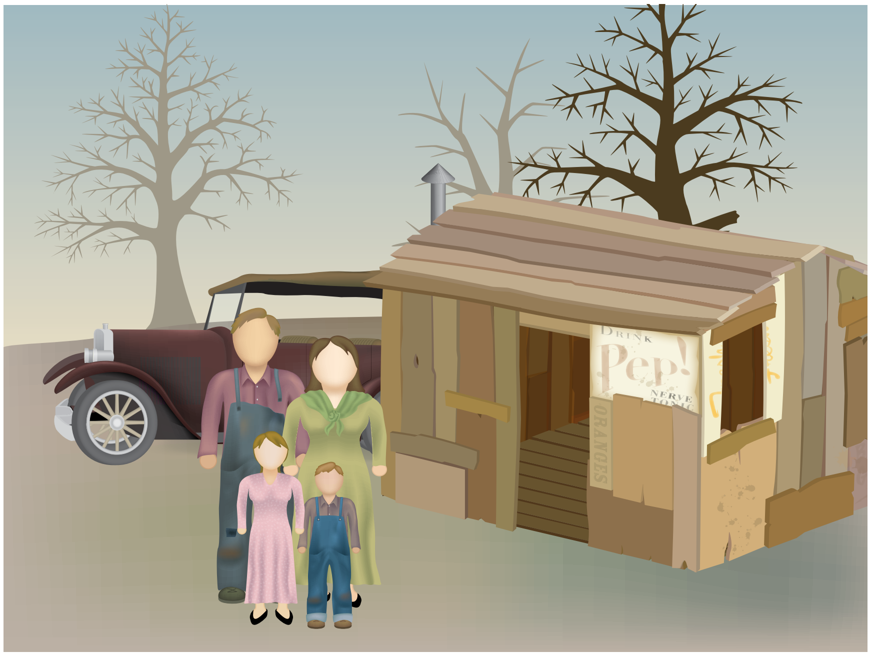
Characters within the project



Some vehicles included in project

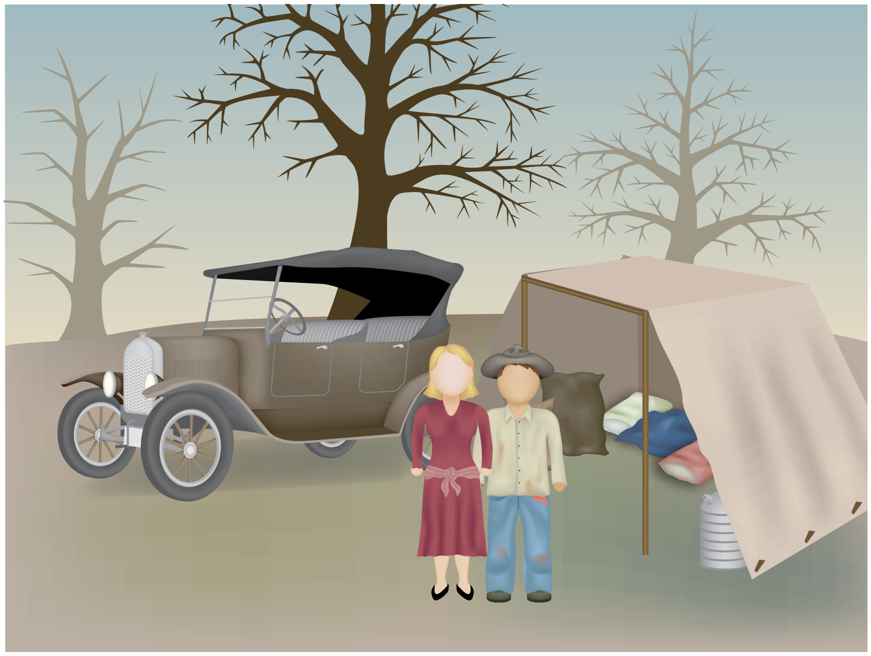


Closeup of tent scene

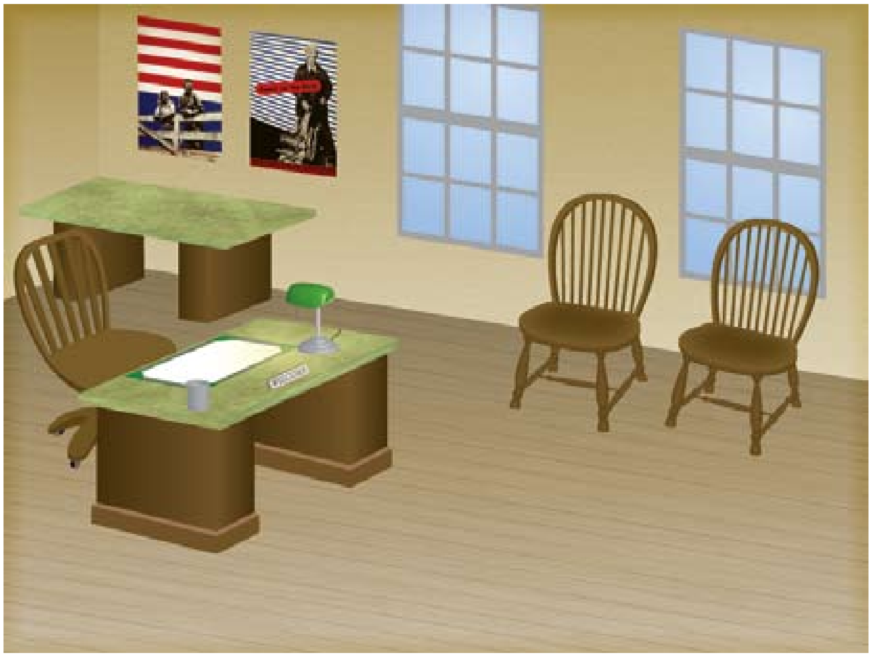


Graphic Development Process:

Additional Screens continued –Closeup of tent scene



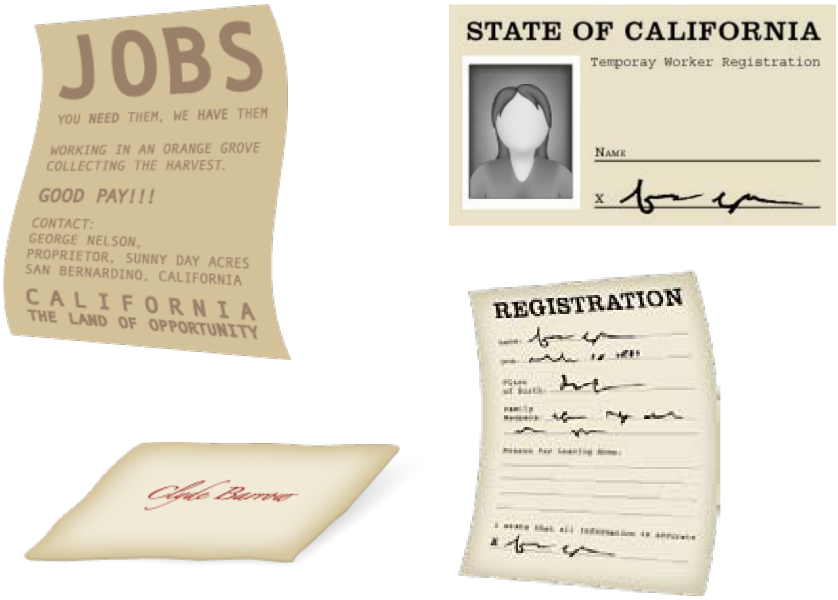
Community Center



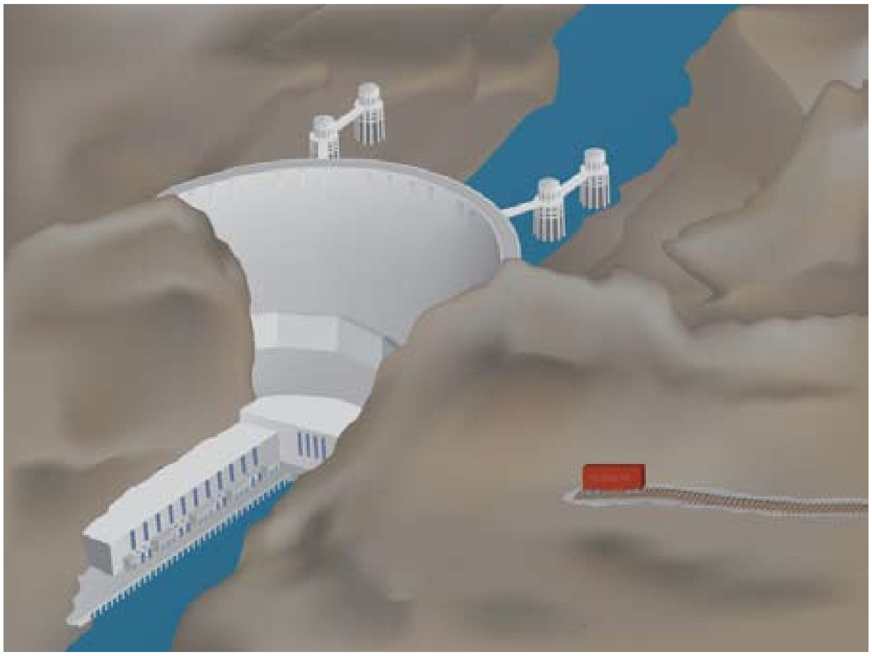
Migration During the Dust Bowl

Graphic Development Process:

Items collected along the way



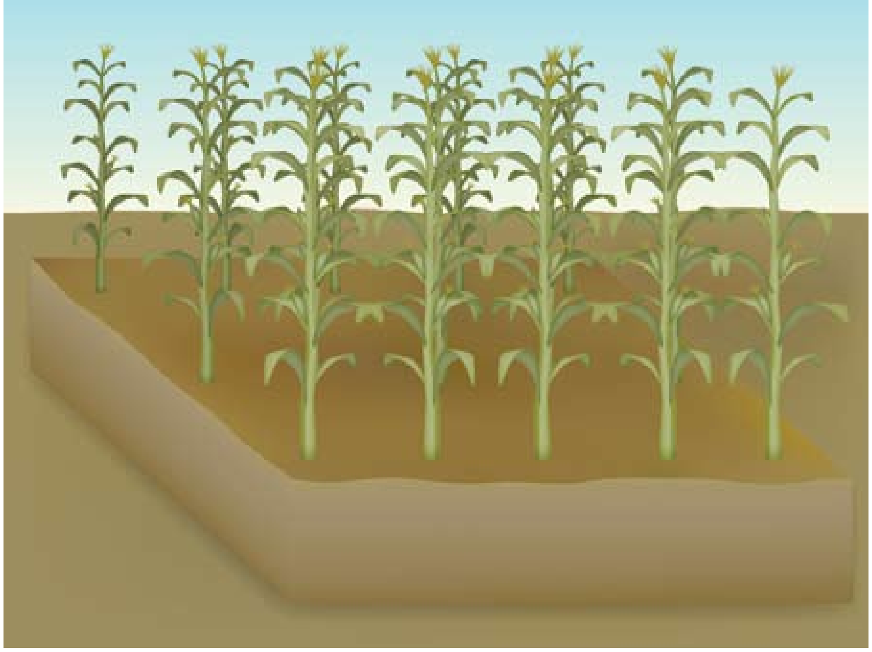
Screens and content developed for interactions – Hoover Dam



Graphic Development Process:

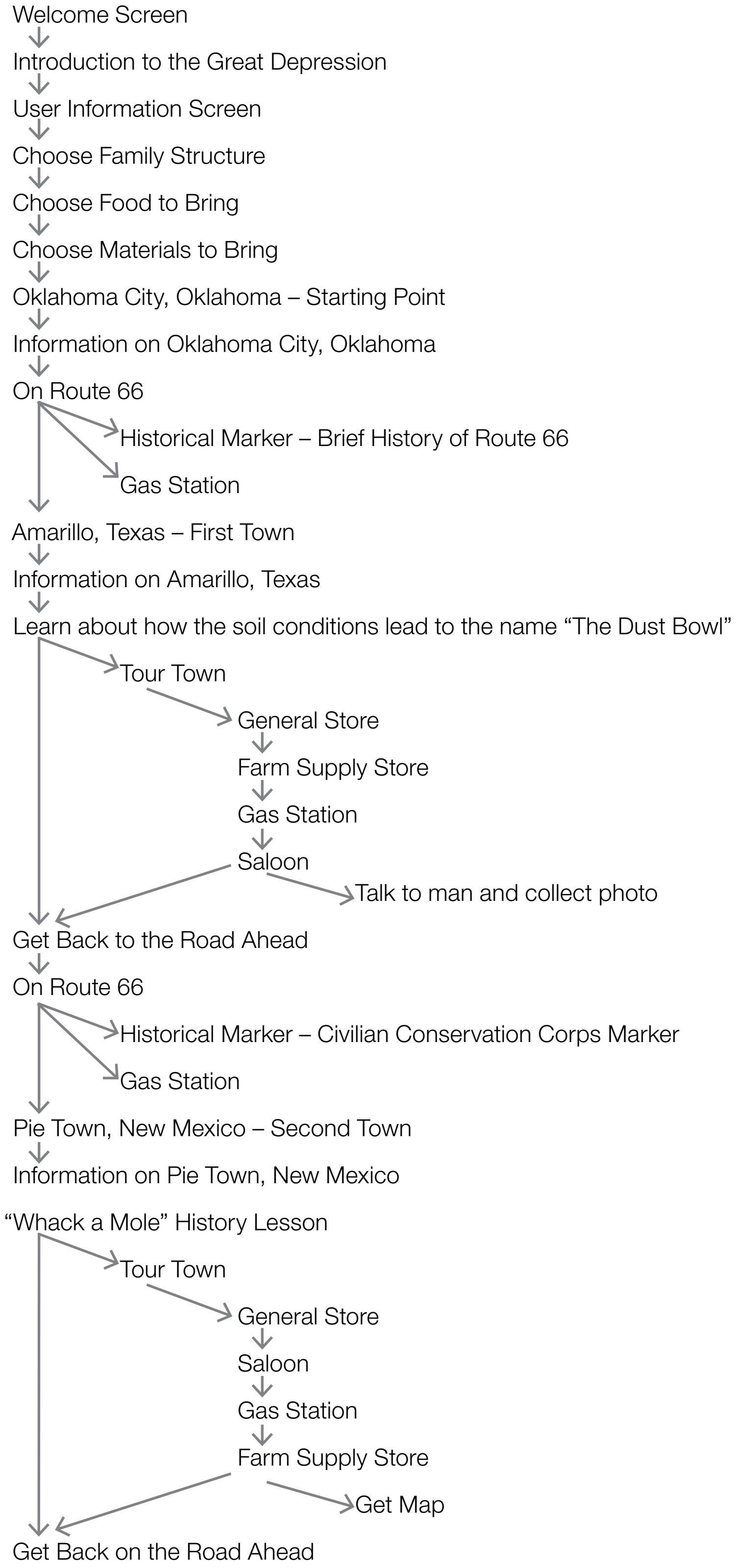
Screens and content developed for interactions – Whack-A-Mole

Screens and content developed for interactions –Soil Strata Examples



Migration During the Dust Bowl

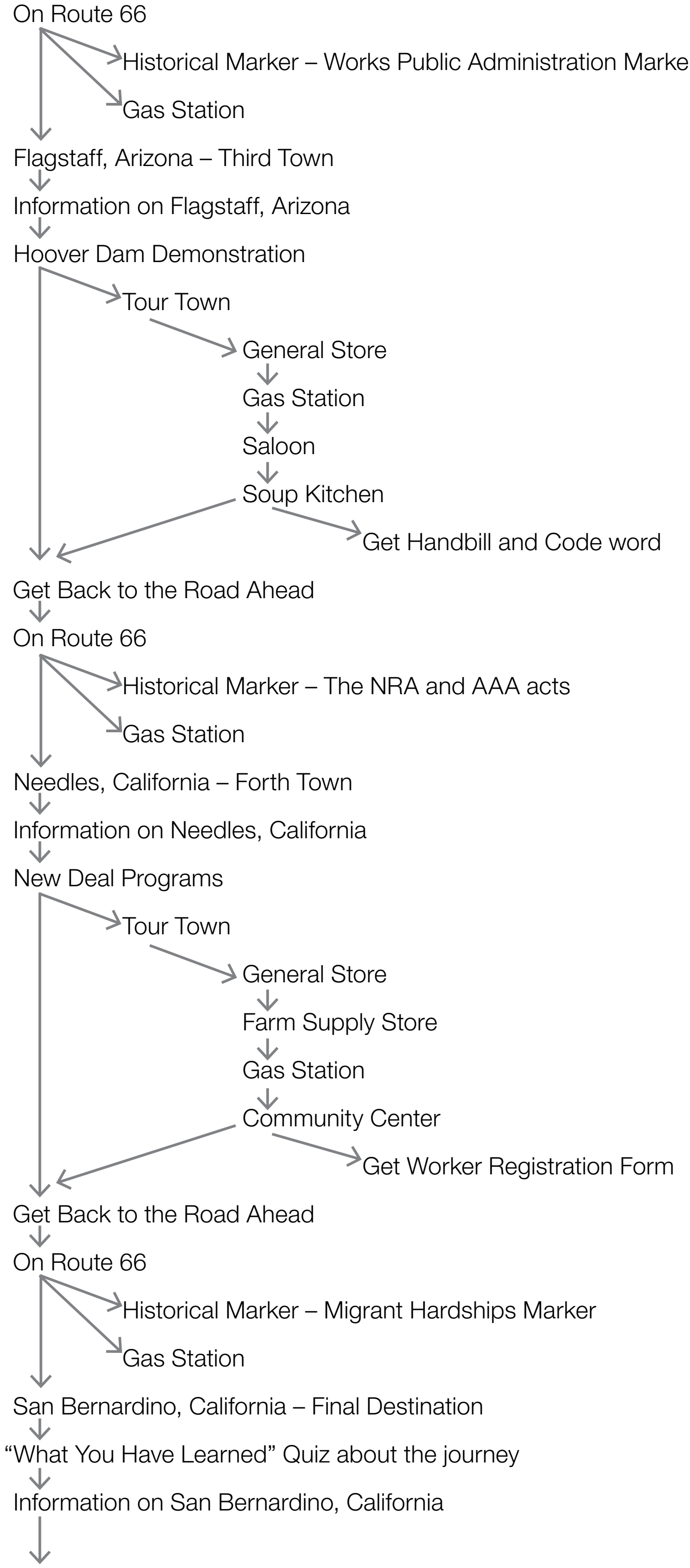
Proposed Flowchart



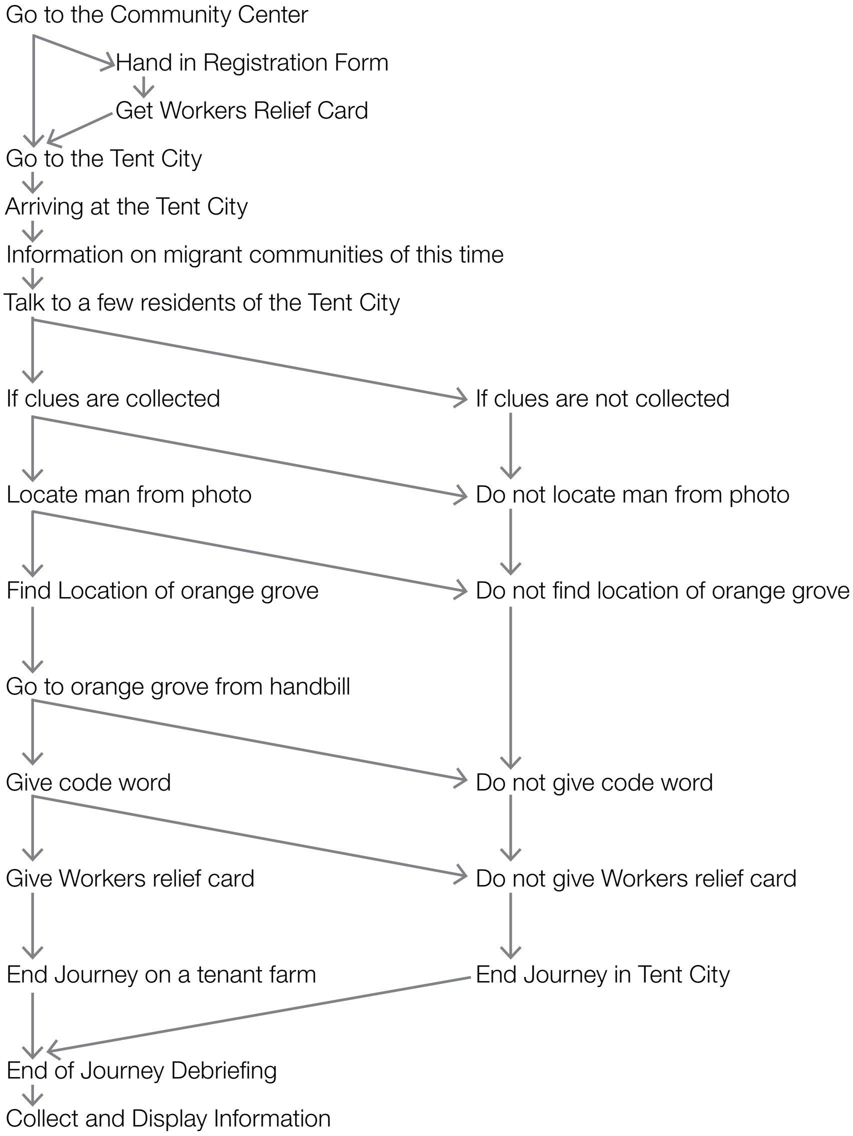
Welcome Screen
Introduction to the Great Depression
User Information Screen
Choose Family Structure
Choose Food to Bring
Choose Materials to Bring
Oklahoma City, Oklahoma – Starting Point
Information on Oklahoma City, Oklahoma
On Route 66
Historical Marker – Brief History of Route 66
Gas Station
Amarillo, Texas – First Town
Information on Amarillo, Texas
Learn about how the soil conditions lead to the name “The Dust Bowl”
Tour Town
General Store
Farm Supply Store
Gas Station
Saloon
Talk to man and collect photo
Get Back to the Road Ahead
On Route 66
Historical Marker – Civilian Conservation Corps Marker
Gas Station
Pie Town, New Mexico – Second Town
Information on Pie Town, New Mexico
“Whack a Mole” History Lesson
Tour Town
General Store
Saloon
Gas Station
Farm Supply Store
Get Map
Get Back on the Road Ahead

(Continued on next page)

Proposed Flowchart (Continued)


On Route 66
Historical Marker – Works Public Administration Marker:
Gas Station
Flagstaff, Arizona – Third Town
Information on Flagstaff, Arizona
Hoover Dam Demonstration
Tour Town
General Store
Gas Station
Saloon
Soup Kitchen
Get Handbill and Code word
Get Back to the Road Ahead
On Route 66
Historical Marker – The NRA and AAA acts
Gas Station
Needles, California – Forth Town
Information on Needles, California
New Deal Programs
Tour Town
General Store
Farm Supply Store
Gas Station
Community Center
Get Worker Registration Form
Get Back to the Road Ahead
On Route 66
Historical Marker – Migrant Hardships Marker
Gas Station
San Bernardino, California – Final Destination
“What You Have Learned” Quiz about the journey
Information on San Bernardino, California

Proposed Flowchart (Continued)


Go to the Community Center
Hand in Registration Form
Get Workers Relief Card
Go to the Tent City
Arriving at the Tent City
Information on migrant communities of this time
Talk to a few residents of the Tent City
If clues are collected
If clues are not collected
Locate man from photo
Do not locate man from photo
Find Location of orange grove
Do not find location of orange grove
Go to orange grove from handbill
Give code word
Do not give code word
Give Workers relief card
Do not give Workers relief card
End Journey on a tenant farm
End Journey in Tent City
End of Journey Debriefing
Collect and Display Information

Migration During the Dust Bowl

Lesson:	Interactive soil strata demonstration
Rationale for Lesson:	During the Dust Bowl, improper use and management of agricultural lands only compounded the drought situation in the Midwest region of the United States, causing the failing crop yields which led to farmers abandoning their fields in search of work.
Teaching Methods & Strategies:	Scientific Method of Inquiry
Expected Learning Outcomes:	At the completion of this interactive challenge, students will develop a better understanding of how soil mismanagement, compounded with drought played a large role in creating the conditions of the Dust Bowl
Evaluation Process:	Through situation gaming, students will be encouraged to use drag and drop interactions to correctly build the soil layers in their correct sequence. Incorrect placement of soil layers will not give the desired result. By correctly identifying the cause for soil erosion, students will be able to progress to the next step of the game.
Materials/Graphics Necessary:	Soil Strata demonstration illustrating the different soil layers

Migration During the Dust Bowl

Goals: At the end of this interactive section, students will know:	**Objectives:** At the end of this interactive section, students will :	**Strategies:** How students will learn about the goals and objectives:
How the improper management of the soil created the effects of the Dust Bowl.	Identify specific causes for the Dust Bowl conditions.	Students will observe a demonstration on why the soil in the Mid West region failed. Imagery and first hand accounts will demonstrate how massive dust storms could be.
	Identify crops which caused the largest amount of soil nutrient and moisture depletion.	Through trial and error processes, learn how the soil was constructed, and how this had an effect on the Dust Bowl. Examples will show which crops demanded the most nutrients and water for successful growth, as well as identify viable alternatives.
Learn the effects the decline in agricultural production had on the American Economy	Assess the effects drought had in the decline in crop yields.	Students will learn how the depressed prices of agricultural goods affected farmers lives. Learn the history of how long the drought took place in the Midwest.
	Construct arguments explaining the need for agricultural regulation.	Explain how unregulated prices affected the cost of crops.

Migration During the Dust Bowl

Lesson:	Interactive Historical Figures Challenge
Rationale for Lesson:	During the Great Depression, the American populous was paralyzed under the grip of little to no monetary savings and a failing economy. By 1933, 25 percent of the population was unemployed and looking for any job that could help sustain their families. They looked towards anyone who could help them. At this time, individuals stood up to the challenges and tried to best shape policies and programs that would best meet these problems. This lesson will focus on these individuals and highlight their accomplishments.
Teaching Methods & Strategies:	Multiple Choice
Expected Learning Outcomes:	At the completion of this interactive challenge, the student will know key historical figures, and how their actions affected the policies and ideas with the United States at the time.
Evaluation Process:	Through questions with multiple choices, students will associate the key figure with the fact. If the student chooses the wrong figure, an action will indicate an incorrect choice, and the student will have an opportunity to choose another. By associating the correct figures to their place in history, students will be able to progress to the next step of the game.
Materials/Graphics Necessary:	Interface containing three holes, where historical figures will emerge. About 12-15 images of historical figure head to cycle randomly through the game.

Migration During the Dust Bowl

Goals: At the end of this interactive section, students will know:	**Objectives:** At the end of this interactive section, students will :	**Strategies:** How students will learn about the goals and objectives:
Learn the major figures of the time period, and how they influenced public policy.	Analyze major figures, and their policies for and against governmental action.	Imagery and brief biographies will introduce the major figures. Questions relevant to specific figures that students will have to correctly match to the individual.
Learn about policies enacted or rejected on how to solve the dire economic and social problems of the time.	Evaluate major policy decisions of the time, and synthesize how they could effect the country's direction.	Introduce major policies of the time and the goals they sought to achieve. Questions relevant to specific policies that students will have to correctly match to the individual.

Migration During the Dust Bowl

Lesson:

Interactive Boulder Dam (Hoover Dam) puzzle

Rationale for Lesson:

A large amount of public works projects were undertaken during the Depression era. A source of employment for the population of the United States, as well as having the added benefit of improving society, transportation and raising the standard of living for many Americans. A demonstration how one large project, such as the Boulder Dam, would not only show how the massive structure was undertaken, how many people were under employ for the project, but would also show the regulation of the water supply from the Colorado river improved irrigation for farms in the Southwest and how it provided power to many communities that had not been introduced to this technology at the time.

Teaching Methods & Strategies:

Item Sequencing
Simulation Gaming

Expected Learning Outcomes:

At the completion of this interactive challenge, the student will develop an understanding that public works projects contain many more layers in terms of affecting society than providing water or electricity or related benefits. As a major employer of workers, it provided cash infusions to the communities near to the project; as well as a sense of well being and accomplishment for being able to create structures that change society as a whole.

Evaluation Process:

By correctly assembling the pieces of the Dam, the students wills see the results of the process.

Through experimentation and simulating the engineer on the project, learn the correct way to assemble such structures.

By sequencing the puzzle correctly, students will be able to progress to the next step of the game.

Materials/Graphics Necessary:

Boulder Dam graphic broken into it’s component parts.
Rendering of Black Canyon where the dam is located
Area Map

Migration During the Dust Bowl

Goals: At the end of this interactive section, students will know:	**Objectives:** At the end of this interactive section, students will :	**Strategies:** How students will learn about the goals and objectives:
How public works programs aimed to improve economic and social conditions for Americans.	Understand the key problems that caused social and economic hardships faced by Americans.	Through a history lesson that corresponds to the content provided. Imagery and first hand accounts of workers constructing the Boulder Dam
	Identify the 2 programs developed by the government to help employ the working class.	Through descriptions of the government programs
Through completion of the Boulder Dam puzzle, students will gain knowledge on the proper methods for building public works projects.	Arrange dam puzzle sections in order to complete the assignment.	Through item sequencing, students will grasp the knowledge of building public works projects. Imagery and first hand accounts will demonstrate how massive dust storms could be.

Migration During the Dust Bowl

Lesson:	Exploration of Famous Chroniclers of the Time
Rationale for Lesson:	During the Dust Bowl, the mediums of large mural painting, music and photography came to being as a way to chronicle the issues of the time. Famous artist like Woody Guthrie, Dorothea Lange and Walker Evans delved deep into the social and cultural issues faced by the public, and found ways of expressing these issues in mediums that created sensitivity among the general public. Through the display of these images, I hope to effect the same feelings of empathy and social concern.
Teaching Methods & Strategies:	Interactive Interview Online discussion?
Expected Learning Outcomes:	At the completion of this interactive challenge, students will develop a better understanding of the societal impacts the Great Depression had on the lives of millions of Americans.
Evaluation Process:	Through an interactive interview with either Dorothea Lange or Woody Guthrie, learn why they chose to chronicle the times through their chosen mediums. Through viewing photographs and audio clips, students will be able to identify the common themes of the era. This will enable the student to more accurately empathize with the struggles everyday Americans faced.
Materials/Graphics Necessary:	Music Clips, and Images of artists contemporary to the period. Interface for students to interact with.

Migration During the Dust Bowl

Goals: At the end of this interactive section, students will know:	**Objectives:** At the end of this interactive section, students will :	**Strategies:** How students will learn about the goals and objectives:
The societal impacts of a depressed economy.	Identify with individuals that faced hardships due to the Great Depression.	Imagery and first hand accounts will demonstrate conditions faced by Americans of the time. Music from individuals like Woody Guthrie will strengthen the material presented. Students will analyze the photos and music for common themes. Through photographs and audio clips, the student will enable different forms of learning techniques.
Famous individuals who documented the conditions during the Great Depression.	Comprehend why famous artists chose to document the time.	Through interviews of these individuals, learn their ideas behind the works of art they created. Analysis of the artist's mediums and subject areas will develop insight into the plight of the era.

Actionscript Code: Frame 1, Layer “Extra Functions”

```
stop();
bytesLoaded = 0;
bytesTotal = _root.getBytesTotal();
_root.onEnterFrame = function() {
    // if there is more to load
    if (bytesLoaded<bytesTotal) {
        // get current amount loaded
        bytesLoaded = _root.getBytesLoaded();
        // calculate percentage
        percentLoaded = Math.round(100*bytesLoaded/bytesTotal);
        // if there is still more
        if (bytesLoaded<bytesTotal) {
            // display text
            //if (percentLoaded<=50) {
            //    this.loader_mc.loadDescription_txt.text = “interface”;
            //}
            //if (percentLoaded>=50) {
            //    this.loader_mc.loadDescription_txt.text = “xml components”;
            //}
            this.loader_mc.loadDescription_txt.text = “interface”;
            this.loader_mc.loading_txt.text = “loading”;
            this.loader_mc.percent_txt.text = percentLoaded+”%”;
            // set scale of bar
            this.loader_mc.bars_mc._x = (percentLoaded*2)-200;
            // no more left
        } else {
            // display complete
            //this.loader_mc.loading_txt.text = “loading”;
            //this.loader_mc.loadDescription_txt.text = “complete”;
            // fill out bar
            this.loader_mc.bars_mc._x = 200;
            // go to next frame
            _root.gotoAndStop(2);
        }
    }
};
```

Actionscript Code: Frame 2, Layer “Extra Functions”

```
getGas = function () {
    _root.movieClipLoader_mc.currentMC.tripViewer_mc.routeGas_mc.onEnterFrame = function() {
        if (puttputt == true) {
            this._x += 15;
            if (this._x>=_root.movieClipLoader_mc.currentMC.tripViewer_mc.modelT_mc._x-30 && this._x<=_root.movieClipLoader_
            mc.currentMC.tripViewer_mc.modelT_mc._x-16) {
                puttputt = false;
                soundFX(“BUZZER”);
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.attachMovie(“wantGas_mc”, “wantGas_mc”, 100);
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc._x = 100;
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc._y = 100;
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc.want_txt.text = “want gas?”;
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc.gasPrice_txt.text = “$0.”+gasPrice();
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc.yes_mc.onPress = function() {
                    addGas();
                    soundFX(“POP”);
                    puttputt = true;
                    _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc.unloadMovie();
                };
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc.no_mc.onPress = function() {
                    puttputt = true;
                    soundFX(“POP”);
                    _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantGas_mc.unloadMovie();
                };
            }
            if (this._x>=875) {
                this._x = -275;
                delete this.onEnterFrame;
            }
        }
    };
};
readMarker = function (mileMarker) {
    _root.movieClipLoader_mc.currentMC.tripViewer_mc.marker_mc.onEnterFrame = function() {
        if (puttputt == true) {
            this._x += 15;
            if (this._x>=_root.movieClipLoader_mc.currentMC.tripViewer_mc.modelT_mc._x-30 && this._x<=_root.movieClipLoader_
            mc.currentMC.tripViewer_mc.modelT_mc._x-16) {
                puttputt = false;
                soundFX(“BUZZER”);
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.attachMovie(“wantGas_mc”, “wantMarker_mc”, 100);
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantMarker_mc._x = 100;
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantMarker_mc._y = 100;
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantMarker_mc.want_txt.text = “read marker?”;
                //
                if (mileMarker == 80) {
                    myMarkerTitle = allKnowing.allMarkers[0].attributes.item;
                    myMarkerInfo = allKnowing.allMarkers[0].attributes.info;
                } else if (mileMarker == 510) {
                    myMarkerTitle = allKnowing.allMarkers[1].attributes.item;
                    myMarkerInfo = allKnowing.allMarkers[1].attributes.info;
                } else if (mileMarker == 775) {
                    myMarkerTitle = allKnowing.allMarkers[2].attributes.item;
                    myMarkerInfo = allKnowing.allMarkers[2].attributes.info;
                } else if (mileMarker == 1050) {
                    myMarkerTitle = allKnowing.allMarkers[3].attributes.item;
                    myMarkerInfo = allKnowing.allMarkers[3].attributes.info;
                } else if (mileMarker == 1220) {
                    myMarkerTitle = allKnowing.allMarkers[4].attributes.item;
                    myMarkerInfo = allKnowing.allMarkers[4].attributes.info;
                }
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantMarker_mc.yes_mc.onPress = function() {
                    gameMarkerArray.push(“yes”);
                    soundFX(“POP”);
                    _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantMarker_mc.unloadMovie();
                    _root.createEmptyMovieClip(“townInfo”, 1001);
                    _root.townInfo.attachMovie(“townInfoMC”, “townInfo_mc”, 1);
                    _root.townInfo.townInfo_mc._x = 200;
                    _root.townInfo.townInfo_mc._y = 300;
                    alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 2, _root.townInfo.townInfo_mc);
                    _root.townInfo.townInfo_mc.townName_txt.text = myMarkerTitle;
                    _root.townInfo.townInfo_mc.townInfo_txt.text = myMarkerInfo;
                    _root.townInfo.townInfo_mc.close_btn.onPress = function() {
                        soundFX(“POP”);
                        _root.townInfo.unloadMovie();
                        puttputt = true;
                    };
                };
                _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantMarker_mc.no_mc.onPress = function() {
                    gameMarkerArray.push(“no”);
                    puttputt = true;
                    soundFX(“POP”);
                    _root.movieClipLoader_mc.currentMC.tripViewer_mc.wantMarker_mc.unloadMovie();
                };
            }
            if (this._x>=800) {
                this._x = -275;
                delete this.onEnterFrame;
            }
        }
    };
};
calculateSupplies = function () {
    newDay = int(newDay)+100;
    dayCount++;
    checkFood();
    checkWeight();
    calculateGas();
};
calculateGas = function () {
    if (totalWeight<50) {
        gasMileage = 15;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=50 && totalWeight<60) {
        gasMileage = 14;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=60 && totalWeight<70) {
        gasMileage = 13;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=70 && totalWeight<80) {
        gasMileage = 12;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=80 && totalWeight<90) {
        gasMileage = 11;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=90 && totalWeight<100) {
        gasMileage = 10;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=100 && totalWeight<110) {
        gasMileage = 9;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=110 && totalWeight<120) {
        gasMileage = 8;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=120 && totalWeight<125) {
        gasMileage = 7;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
    if (totalWeight>=125) {
        gasMileage = 5;
        return totalGas=int(gasMileage)*int(gasGallons);
    }
```

```
};
addMiles = function () {
        //totalGas -= 1;
        mileCounter += .5;
        _root.movieClipLoader_mc.currentMC.miles_txt.text = mileCounter;
        gasGallons = gasGallons-(.25/gasMileage);
        _root.informationMC.interface_mc.gasGauge_mc.needle_mc._rotation = int(gasMileage/10+gasGallons*10)-55;
        //_root.movieClipLoader_mc.currentMC.gallons_txt.text = gasGallons;
        //_root.movieClipLoader_mc.currentMC.gallons_txt.maxChars = 4;
        if (gasGallons<=0) {
                attachMessage(“gas”, “You’ve run out of gas! you’ll have to spend a day to get gas”);
        }
        //_root.movieClipLoader_mc.currentMC.mpg_txt.text = gasMileage;
        //_root.movieClipLoader_mc.currentMC.gas_txt.text = totalGas;
        //_root.movieClipLoader_mc.currentMC.food_txt.text = foodWeight;
        //checkWeight(“food”);
        //totalWeight = foodWeight+supplyWeight;
        //_root.movieClipLoader_mc.currentMC.weight_txt.text = totalWeight;
        // checkWeight(“total”);
        //_root.movieClipLoader_mc.currentMC.cash_txt.text = “$”+totalCash;
        //_root.movieClipLoader_mc.currentMC.cash_txt.maxChars = 4;
};
addGas = function () {
        gasNeeded = Math.abs(10-gasGallons);
        buyGas = (myGasPrice*gasNeeded)*.01;
        trace(“how much gas to buy? “+buyGas);
        trace(“how much gas needed? “+gasNeeded);
        trace(“how much gas to cost? “+myGasPrice);
        totalCash = totalCash-buyGas;
        gasGallons = 10;
        calculateGas();
        _root.informationMC.interface_mc.gasGauge_mc.needle_mc._rotation = int(gasMileage/10+gasGallons*10)-55;
        //_root.movieClipLoader_mc.currentMC.gas_txt.text = totalGas;
};
gasPrice = function () {
        var myMinPrice = allKnowing.allPrices[1].attributes.min;
        var myMaxPrice = allKnowing.allPrices[1].attributes.max;
        return myGasPrice=int(myRandom(int(myMinPrice), int(myMaxPrice)));
};
deathCounter = function () {
        if (puttputt == true) {
                if (noFood == true) {
                        var myDeathNum = int(myRandom(8, 500000));
                        if (myDeathNum<=12) {
                                var myMemNum = int(myRandom(0, gamePlayerArray.length-1));
                                var q = int(myDeathNum);
                                var myMember = gamePlayerArray[myMemNum];
                                trace(myMember);
                                var penalty1 = myMember+” “+allKnowing.allCosts[q].attributes.item+”. “;
                                var penalty2 = “You’ll have to spend 1 day from your travels. “;
                                var penaltyPrice = allKnowing.allCosts[q].attributes.price;
                                var penaltyDeath = allKnowing.allCosts[q].attributes.death;
                                puttputt = false;
                                attachMessage(“death”, penalty1+penalty2, penaltyDeath, myMemNum);
                        }
                }
        } else if (puttputt == false) {
                null;
        }
};
weightCounter = function () {
    if (puttputt == true) {
        if (overweight == true) {
            var myPenaltyNum = int(myRandom(0, 50000));
            if (myPenaltyNum<=7) {
                var q = int(myPenaltyNum);
                var penalty1 = “You’ve gotten a “+allKnowing.allCosts[q].attributes.item+”. “;
                var penalty2 = “It’ll cost you “+allKnowing.allCosts[q].attributes.price+” to fix, “;
                var penalty3 = “and you’ll have to spend “+allKnowing.allCosts[q].attributes.penalty+” days from your trip”;
                var penaltyPrice = allKnowing.allCosts[q].attributes.price;
                var penaltyDay = allKnowing.allCosts[q].attributes.penalty;
                puttputt = false;
                attachMessage(“penalty”, penalty1+penalty2+penalty3, penaltyPrice, penaltyDay);
            }
        }
    } else if (puttputt == false) {
        null;
    }
};
penaltyCounter = function () {
    if (puttputt == true) {
        var myPenaltyNum = int(myRandom(0, 50000));
        if (myPenaltyNum<=7) {
            var q = int(myPenaltyNum);
            var penalty1 = “You’ve gotten a “+allKnowing.allCosts[q].attributes.item+”. “;
            var penalty2 = “It’ll cost you “+allKnowing.allCosts[q].attributes.price+” to fix, “;
            var penalty3 = “and you’ll have to spend “+allKnowing.allCosts[q].attributes.penalty+” days from your trip”;
            var penaltyPrice = allKnowing.allCosts[q].attributes.price;
            var penaltyDay = allKnowing.allCosts[q].attributes.penalty;
            puttputt = false;
            attachMessage(“penalty”, penalty1+penalty2+penalty3, penaltyPrice, penaltyDay);
        }
        if (myPenaltyNum>8 && myPenaltyNum<=12) {
            var myMemNum = int(myRandom(0, gamePlayerArray.length-1));
            var q = int(myPenaltyNum);
            var myMember = gamePlayerArray[myMemNum];
            trace(myMember);
            var penalty1 = myMember+” “+allKnowing.allCosts[q].attributes.item+”. “;
            var penalty2 = “You’ll have to spend 1 day from your travels. “;
            var penaltyPrice = allKnowing.allCosts[q].attributes.price;
            var penaltyDeath = allKnowing.allCosts[q].attributes.death;
            puttputt = false;
            attachMessage(“death”, penalty1+penalty2, penaltyDeath, myMemNum);
        }
    } else if (puttputt == false) {
        null;
    }
};
attachMessage = function (action, words, cost1, cost2) {
    puttputt = false;
    checkTotalMovement(puttputt);
    _root.movieClipLoader_mc.currentMC.attachMovie(“messageMC”, “messageMC”, 100);
    _root.movieClipLoader_mc.currentMC.messageMC.text1_txt.text = words.toString();
    _root.movieClipLoader_mc.currentMC.messageMC.submit_mc.onPress = function() {
        if (action == “gas”) {
            trace(“gas!”);
            penaltyCount++;
            calculateSupplies();
            addGas();
            soundFX(“POP”);
            _root.movieClipLoader_mc.currentMC.messageMC.unloadMovie();
            puttputt = true;
        } else if (action == “penalty”) {
            penaltyCount++;
            happiness--;
            trace(“penalty!”);
            soundFX(“POP”);
            totalCash = totalCash-cost1;
            if (cost2 == “1”) {
                calculateSupplies();
            }
            _root.movieClipLoader_mc.currentMC.messageMC.unloadMovie();
            puttputt = true;
        } else if (action == “death”) {
            penaltyCount++;
            happiness--;
            trace(“death!”);
            soundFX(“POP”);
            calculateSupplies();
            if (cost1 == “1”) {
                happiness--;
                gamePlayerArray.splice(int(cost2), int(cost2)+1);
                trace(gamePlayerArray);
            }
            _root.movieClipLoader_mc.currentMC.messageMC.unloadMovie();
            puttputt = true;
        } else if (action == “food”) {
            _root.movieClipLoader_mc.currentMC.messageMC.unloadMovie();
            soundFX(“POP”);
            puttputt = true;
        } else if (action == “weight”) {
            _root.movieClipLoader_mc.currentMC.messageMC.unloadMovie();
            soundFX(“POP”);
            puttputt = true;
        }
    };
};
attachRules = function (inGame) {
    puttputt = false;
    _root.createEmptyMovieClip(“rulesMC”, 98765);
    _root.rulesMC.attachMovie(“gameRules_mc”, “gameRules_mc”, 1);
    alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 1, _root.rulesMC.gameRules_mc);
    _root.rulesMC.gameRules_mc._x = Stage.width/2;
    _root.rulesMC.gameRules_mc._y = Stage.height/2;
    //
    _root.rulesMC.gameRules_mc.gotoAndStop(inGame);
    //
    _root.rulesMC.gameRules_mc.close_btn.onPress = function() {
        _root.rulesMC.unloadMovie();
        soundFX(“POP”);
        puttputt = true;
    };
};
attachStory = function () {
    _root.createEmptyMovieClip(“storyMC”, 98764);
    _root.storyMC.attachMovie(“theStory_mc”, “theStory_mc”, 1);
    alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 1, _root.storyMC.theStory_mc);
    _root.storyMC.theStory_mc._x = Stage.width/2;
    _root.storyMC.theStory_mc._y = Stage.height/2;
    //
    _root.storyMC.theStory_mc.gotoAndStop(1)
    //
    _root.storyMC.theStory_mc.close_btn.onPress = function() {
        _root.storyMC.unloadMovie();
        soundFX(“POP”);
    };
};
attachClues = function (thisClue) {
    trace(thisClue);
    if (thisClue == “unload”) {
        trace(“unloading clue”);
        _root.myClue_mc.unloadMovie();
    } else {
        trace(“attaching clue”);
        clueWorks = “myClue_mc”;
        clueTest = _root.createEmptyMovieClip(clueWorks, 12345);
        gridClip = _root[clueWorks].attachMovie(thisClue, “thisClue_mc”, 1);
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 1, clueTest);
        clueTest._x = Stage.width/2;
        clueTest._y = Stage.height/2;
    }
};
pushClues = function () {
    gameCluesArray.push(gameClue);
    trace(“my clues are: “+gameCluesArray);
    gameClue = false;
};
checkClues = function (getAClue) {
    clueValue = gameCluesArray[getAClue];
    //trace(“my clues are: “+gameCluesArray);
    //trace(“my clues are: “+clueValue);
    return clueValue;
};
clearTalk = function () {
    _root.talk_mc.talkScreen_mc._y = -1000;
};
soundFX = function (mySound) {
    newFX = new Sound(this);
    newFX.attachSound(mySound);
    if (mySound == “EXIT”) {
        newFX.setVolume(40);
    } else {
        newFX.setVolume(15);
    }
    newFX.start(0, 0);
};
```

Actionscript Code: Frame 2, Layer “Frame Actions”

```
stop();
// tracking XML vars
_global.allKnowing = new Object();
// tracking loaded clips
frameCount = 1;
// tracking travel and gas
puttputt = true;
newDay = 100;
dayCount = 1;
penaltyCount = 0;
mileCounter = 0;
gasGallons = 10;
gasMileage = 10;
// tracking food, supplies and cash
```

```
totalWeight = 0;
foodWeight = 0;
totalCash = 50;
//int(allKnowing.allPrices[0].attributes.price);
supplyWeight = 0;
_global.foodSupplies = new Object();
_global.totalSupplies = new Object();
// tracking family members
gamePlayerArray = new Array();
gameMemberArray = new Array();
gameGenderArray = new Array();
familyArray = new Array();
// tracking game goals
gameClue = false;
gameMarkerArray = new Array();
gameCluesArray = new Array();
// assorted variables
incurredPenalty = false;
happiness = 10;
noFood = false;
overweight = false;
foodCounter = false;
supplyCounter = false;
// all vars should be set to XML nodes for updatability on the fly...
//
// add any movie clips for content
_root.createEmptyMovieClip(“informationMC”, 50);
//
_root.createEmptyMovieClip(“desc_mc”, 35791);
_root.desc_mc.attachMovie(“itemDescriptionMC”, “itemDescription_mc”, 498);
_root.desc_mc.itemDescription_mc._y = -1000;
_root.createEmptyMovieClip(“talk_mc”, 36000);
_root.talk_mc.attachMovie(“talkScreenMC”, “talkScreen_mc”, 498);
_root.talk_mc.talkScreen_mc._y = -1000;
_root.createEmptyMovieClip(“speak_mc”, 36237);
_root.speak_mc.attachMovie(“speakToMe”, “speakMC”, 498);
_root.speak_mc.speakMC._y = -1000;
//
_root.advance_mc.onRelease = function() {
    this._alpha = 0;
    if (frameCount<allKnowing.myFiles-1) {
        frameCount++;
        loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
    }
};
loadFrame = function (mcName, mcLoad) {
    this.movieClipLoader_mc.attachMovie(mcLoad, “currentMC”, 1);
    //this.gameLoc_txt.text = mcName;
    loadedMovie = mcLoad;
    checkContent();
};
checkContent = function () {
    trace(frameCount);
    if (frameCount == 0) {
        _root.movieClipLoader_mc.currentMC.attachMovie(“exit_mc”, “exit_mc”, 2);
        _root.movieClipLoader_mc.currentMC.exit_mc._x = 240;
        _root.movieClipLoader_mc.currentMC.exit_mc._y = 200;
        _root.movieClipLoader_mc.currentMC.exit_mc.onPress = function() {
            soundFX(“EXIT”);
            frameCount++;
            _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        };
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
    } else if (frameCount == 1) {
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.playGame_mc.onRelease = function() {
            if (frameCount<allKnowing.myFiles-1) {
                frameCount++;
                soundFX(“BUZZER”);
                loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
            }
        };
        _root.movieClipLoader_mc.currentMC.learnStory_mc.onRelease = function() {
            attachStory();
            soundFX(“BUZZER”);
        };
        _root.movieClipLoader_mc.currentMC.gameRules_mc.onRelease = function() {
            attachRules(“start”);
            soundFX(“BUZZER”);
        };
    } else if (frameCount == 2) {
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.informationMC.attachMovie(“interfaceMC”, “interface_mc”, 1);
        _root.informationMC.interface_mc._x = 400;
        _root.informationMC.interface_mc._y = 335;
        _root.informationMC.interface_mc.familyInfo_mc._visible = true;
        _root.informationMC.interface_mc.moneyInfo_mc._visible = false;
        _root.informationMC.interface_mc.foodInfo_mc._visible = false;
        _root.informationMC.interface_mc.supplyInfo_mc._visible = false;
        _root.informationMC.interface_mc.mapInfo_mc._visible = false;
        _root.informationMC.interface_mc.gasGauge_mc._visible = false;
        _root.informationMC.interface_mc.gameInfo_mc._visible = true;
        _root.informationMC.interface_mc.familyInfo_mc;
        var familyMember = 0;
        var dragCount = 0;
        attachRules(“family”);
        allDrag = false;
        _root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“myFather”, 1);
        _root.movieClipLoader_mc.currentMC.myFather.createEmptyMovieClip(“loader_mc”, 1);
        _root.movieClipLoader_mc.currentMC.myFather.loader_mc.attachMovie(“father_mc”, “father”, 1);
        _root.movieClipLoader_mc.currentMC.myFather.loader_mc._xscale = 75;
        _root.movieClipLoader_mc.currentMC.myFather.loader_mc._yscale = 75;
        _root.movieClipLoader_mc.currentMC.myFather._x = -200;
        _root.movieClipLoader_mc.currentMC.myFather._y = 250;
        _root.movieClipLoader_mc.currentMC.myFather.draggable = true;
        _root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“myMother”, 2);
        _root.movieClipLoader_mc.currentMC.myMother.createEmptyMovieClip(“loader_mc”, 1);
        _root.movieClipLoader_mc.currentMC.myMother.loader_mc.attachMovie(“mother_mc”, “mother”, 1);
        _root.movieClipLoader_mc.currentMC.myMother.loader_mc._xscale = 75;
        _root.movieClipLoader_mc.currentMC.myMother.loader_mc._yscale = 75;
        _root.movieClipLoader_mc.currentMC.myMother._x = -150;
        _root.movieClipLoader_mc.currentMC.myMother._y = 250;
        _root.movieClipLoader_mc.currentMC.myMother.draggable = true;
        //
        _root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“brother1”, 4);
        _root.movieClipLoader_mc.currentMC.brother1.createEmptyMovieClip(“loader_mc”, 1);
        _root.movieClipLoader_mc.currentMC.brother1.loader_mc.attachMovie(“son1_mc”, “brother1”, 1);
        _root.movieClipLoader_mc.currentMC.brother1.loader_mc._xscale = 75;
        _root.movieClipLoader_mc.currentMC.brother1.loader_mc._yscale = 75;
        _root.movieClipLoader_mc.currentMC.brother1._x = -250;
        _root.movieClipLoader_mc.currentMC.brother1._y = 250;
        _root.movieClipLoader_mc.currentMC.brother1.draggable = true;
```

```
//
_root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“brother2”, 5);
_root.movieClipLoader_mc.currentMC.brother2.createEmptyMovieClip(“loader_mc”, 1);
_root.movieClipLoader_mc.currentMC.brother2.loader_mc.attachMovie(“son2_mc”, “brother2”, 1);
_root.movieClipLoader_mc.currentMC.brother2.loader_mc._xscale = 75;
_root.movieClipLoader_mc.currentMC.brother2.loader_mc._yscale = 75;
_root.movieClipLoader_mc.currentMC.brother2._x = -200;
_root.movieClipLoader_mc.currentMC.brother2._y = 250;
_root.movieClipLoader_mc.currentMC.brother2.draggable = true;
//
_root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“sister1”, 6);
_root.movieClipLoader_mc.currentMC.sister1.createEmptyMovieClip(“loader_mc”, 1);
_root.movieClipLoader_mc.currentMC.sister1.loader_mc.attachMovie(“daughter1_mc”, “sister1”, 1);
_root.movieClipLoader_mc.currentMC.sister1.loader_mc._xscale = 75;
_root.movieClipLoader_mc.currentMC.sister1.loader_mc._yscale = 75;
_root.movieClipLoader_mc.currentMC.sister1._x = -150;
_root.movieClipLoader_mc.currentMC.sister1._y = 250;
_root.movieClipLoader_mc.currentMC.sister1.draggable = true;
//
_root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“sister2”, 7);
_root.movieClipLoader_mc.currentMC.sister2.createEmptyMovieClip(“loader_mc”, 1);
_root.movieClipLoader_mc.currentMC.sister2.loader_mc.attachMovie(“daughter2_mc”, “sister2”, 1);
_root.movieClipLoader_mc.currentMC.sister2.loader_mc._xscale = 75;
_root.movieClipLoader_mc.currentMC.sister2.loader_mc._yscale = 75;
_root.movieClipLoader_mc.currentMC.sister2._x = -100;
_root.movieClipLoader_mc.currentMC.sister2._y = 250;
_root.movieClipLoader_mc.currentMC.sister2.draggable = true;
//
_root.movieClipLoader_mc.currentMC.nameScreen_mc.name1_txt.restrict = “A-Z a-z”;
//
//familyVisiblity
_root.movieClipLoader_mc.currentMC.myFather._visible = false;
_root.movieClipLoader_mc.currentMC.myMother._visible = false;
_root.movieClipLoader_mc.currentMC.brother1._visible = false;
_root.movieClipLoader_mc.currentMC.brother2._visible = false;
_root.movieClipLoader_mc.currentMC.sister1._visible = false;
_root.movieClipLoader_mc.currentMC.sister2._visible = false;
//
pushName = function (myMC, newFrame) {
        if (myMC.name1_txt.text == null || myMC.name1_txt.text == “”) {
                myMC.error_txt.text = “Please enter a name”;
        } else {
                mainPlayer = myMC.name1_txt.text;
                gamePlayerArray.push(myMC.name1_txt.text);
                myMC.error_txt.text = “”;
                myMC.name1_txt.text = “”;
                myMC.gotoAndStop(newFrame);
                myFamilyStatus();
                allDrag = true;
                //gameMember = true;
        }
};
pushSex = function (sex, myMC, newFrame) {
        gameGenderArray.push(sex);
        allDrag = false;
        //gameMember = false;
        dragCount++;
        //newmember._x = (gameMemberArray[familyMember-1]._x+gameMemberArray[familyMember-1]._width)-20;
        //familyArray[familyMember] = new Array(gamePlayerArray[familyMember], gameGenderArray[familyMember]);
        trace(“MY name : “+gamePlayerArray);
        trace(“MY member : “+gameGenderArray);
        //trace(“MY gender : “+familyArray[familyMember][1]);
        myFamilyStatus();
};
_root.movieClipLoader_mc.currentMC.nameScreen_mc.submit_mc.onRelease = function() {
        pushName(_root.movieClipLoader_mc.currentMC.nameScreen_mc, 2);
};
//
myFamilyStatus = function () {
        familyMember++;
        trace(familyMember);
        trace(“dragged: “+dragCount);
        if (_root.movieClipLoader_mc.currentMC.nameScreen_mc._currentframe == 1) {
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.submit_mc.onRelease = function() {
                        pushName(_root.movieClipLoader_mc.currentMC.nameScreen_mc, 2);
                };
        }
        if (_root.movieClipLoader_mc.currentMC.nameScreen_mc._currentframe == 2) {
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.text1_txt.text = “Please drag the family member over to the gathering
                area”;
        }
        if (familyMember == 1) {
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.submit_mc.onRelease = function() {
                        pushName(_root.movieClipLoader_mc.currentMC.nameScreen_mc, 2);
                };
                _root.movieClipLoader_mc.currentMC.myFather._visible = true;
                _root.movieClipLoader_mc.currentMC.myMother._visible = true;
                //
                _root.movieClipLoader_mc.currentMC.myFather.onPress = function() {
                        if (allDrag == true) {
                                if (this.draggable == true) {
                                        mySex = “father_mc”;
                                        this.swapDepths(10);
                                        startDrag(this, false, -400, -300, 300, 400);
                                }
                        }
                };
                _root.movieClipLoader_mc.currentMC.myFather.onRelease = function() {
                        familyHitArea(this, mySex);
                };
                _root.movieClipLoader_mc.currentMC.myMother.onPress = function() {
                        if (allDrag == true) {
                                if (this.draggable == true) {
                                        mySex = “mother_mc”;
                                        this.swapDepths(10);
                                        startDrag(this, false, -400, -300, 300, 400);
                                }
                        }
                };
                _root.movieClipLoader_mc.currentMC.myMother.onRelease = function() {
                        familyHitArea(this, mySex);
                };
        }
        if (familyMember>=1) {
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.submit_mc.onRelease = function() {
                        pushName(_root.movieClipLoader_mc.currentMC.nameScreen_mc, 2);
                };
        }
        if (familyMember>=5) {
                _root.movieClipLoader_mc.currentMC.brother1._visible = true;
                _root.movieClipLoader_mc.currentMC.brother2._visible = true;
                _root.movieClipLoader_mc.currentMC.sister1._visible = true;
                _root.movieClipLoader_mc.currentMC.sister2._visible = true;
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.text1_txt.text = “Who else is in your family?”;
                //
                //
                _root.movieClipLoader_mc.currentMC.brother1.onPress = function() {
                    if (allDrag == true) {
                        if (this.draggable == true) {
                            mySex = “son1_mc”;
                            this.swapDepths(10);
                            startDrag(this, false, -400, -300, 300, 400);
                        }
                    }
                };
                _root.movieClipLoader_mc.currentMC.brother1.onRelease = function() {
                    familyHitArea(this, mySex);
                };
                _root.movieClipLoader_mc.currentMC.brother2.onPress = function() {
                    if (allDrag == true) {
                        if (this.draggable == true) {
                            mySex = “son2_mc”;
                            this.swapDepths(10);
                            startDrag(this, false, -400, -300, 300, 400);
                        }
                    }
                };
                _root.movieClipLoader_mc.currentMC.brother2.onRelease = function() {
                    familyHitArea(this, mySex);
                };
                _root.movieClipLoader_mc.currentMC.sister1.onPress = function() {
                    if (allDrag == true) {
                        if (this.draggable == true) {
                            mySex = “daughter1_mc”;
                            this.swapDepths(10);
                            startDrag(this, false, -400, -300, 300, 400);
                        }
                    }
                };
                _root.movieClipLoader_mc.currentMC.sister1.onRelease = function() {
                    familyHitArea(this, mySex);
                };
                _root.movieClipLoader_mc.currentMC.sister2.onPress = function() {
                    if (allDrag == true) {
                        if (this.draggable == true) {
                            mySex = “daughter2_mc”;
                            this.swapDepths(10);
                            startDrag(this, false, -400, -300, 300, 400);
                        }
                    }
                };
                _root.movieClipLoader_mc.currentMC.sister2.onRelease = function() {
                    familyHitArea(this, mySex);
                };
            }
        };
        familyHitArea = function (thisMember, mySex) {
            trace(thisMember);
            if (thisMember.hitTest(_root.movieClipLoader_mc.currentMC.family_mc)) {
                thisMember._x = thisMember._x;
                thisMember._y = thisMember._y;
                if (thisMember.draggable == true) {
                    thisMember.draggable = false;
                    pushSex(mySex, _root.movieClipLoader_mc.currentMC.nameScreen_mc, 1);
                }
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.gotoAndStop(1);
            } else {
                thisMember._x = thisMember._x;
                thisMember._y = thisMember._y;
                //drag1 = false;
            }
            thisMember.stopDrag();
            if (familyMember<5) {
                if (mySex == “father_mc”) {
                    _root.movieClipLoader_mc.currentMC.nameScreen_mc.text1_txt.text = “What is your wife’s name?”;
                }
                if (mySex == “mother_mc”) {
                    _root.movieClipLoader_mc.currentMC.nameScreen_mc.text1_txt.text = “What is your husband’s name?”;
                }
            }
            if (familyMember>=5) {
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.onEnterFrame = function() {
                    if (_root.movieClipLoader_mc.currentMC.nameScreen_mc._currentframe == 1) {
                        _root.movieClipLoader_mc.currentMC.nameScreen_mc.text1_txt.text = “Who else is in your family?”;
                    }
                    if (_root.movieClipLoader_mc.currentMC.nameScreen_mc._currentframe == 2) {
                        _root.movieClipLoader_mc.currentMC.nameScreen_mc.text1_txt.text = “Please drag the family member over to the
                        gathering area”;
                    }
                };
            }
            if (dragCount == 5) {
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.gotoAndStop(3);
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.text1_txt.text = “Your family is done. You’re ready to continue...”;
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.error_txt.text = “Press EXIT to continue”;
                if (_root.movieClipLoader_mc.currentMC.sister1.draggable == true) {
                    _root.movieClipLoader_mc.currentMC.sister1.unloadMovie();
                }
                if (_root.movieClipLoader_mc.currentMC.sister2.draggable == true) {
                    _root.movieClipLoader_mc.currentMC.sister2.unloadMovie();
                }
                if (_root.movieClipLoader_mc.currentMC.brother1.draggable == true) {
                    _root.movieClipLoader_mc.currentMC.brother1.unloadMovie();
                }
                if (_root.movieClipLoader_mc.currentMC.brother2.draggable == true) {
                    _root.movieClipLoader_mc.currentMC.brother2.unloadMovie();
                }
                _root.movieClipLoader_mc.currentMC.nameScreen_mc.submit2_mc.onRelease = function() {
                    _root.movieClipLoader_mc.currentMC.myFather.unloadMovie();
                    _root.movieClipLoader_mc.currentMC.myMother.unloadMovie();
                    _root.movieClipLoader_mc.currentMC.brother1.unloadMovie();
                    _root.movieClipLoader_mc.currentMC.brother2.unloadMovie();
                    _root.movieClipLoader_mc.currentMC.sister1.unloadMovie();
                    _root.movieClipLoader_mc.currentMC.sister2.unloadMovie();
                    frameCount++;
                    loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
                };
            }
        };
        myFamilyStatus();
        // Grocery Store
    } else if (frameCount == 3) {
        totalCash = int(allKnowing.allPrices[0].attributes.price);
        _root.informationMC.interface_mc.moneyInfo_mc._visible = true;
```

```
        _root.informationMC.interface_mc.foodInfo_mc._visible = true;
        _root.movieClipLoader_mc.currentMC.attachMovie(“exit_mc”, “exit_mc”, 2);
        _root.movieClipLoader_mc.currentMC.exit_mc._x = 240;
        _root.movieClipLoader_mc.currentMC.exit_mc._y = 200;
        _root.movieClipLoader_mc.currentMC.exit_mc.onPress = function() {
            soundFX(“EXIT”);
            frameCount++;
            _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        };
        //
        foodSupplies.flour = int(myRandom(0, 5));
        foodSupplies.potato = int(myRandom(0, 5));
        foodSupplies.vegetable = int(myRandom(0, 5));
        foodSupplies.pork = int(myRandom(0, 5));
        foodSupplies.sugar = int(myRandom(0, 5));
        foodSupplies.coffee = int(myRandom(0, 5));
        //
        getFood();
        toTalk = “grocery”;
        peopleSpeak(toTalk, 0);
        attachRules(“grocery”);
        //Five & Dime
    } else if (frameCount == 4) {
        trace(totalSupplies);
        _root.informationMC.interface_mc.supplyInfo_mc._visible = true;
        _root.movieClipLoader_mc.currentMC.attachMovie(“exit_mc”, “exit_mc”, 2);
        _root.movieClipLoader_mc.currentMC.exit_mc._x = 240;
        _root.movieClipLoader_mc.currentMC.exit_mc._y = 200;
        _root.movieClipLoader_mc.currentMC.exit_mc.onPress = function() {
            soundFX(“EXIT”);
            frameCount++;
            _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        };
        //
        totalSupplies.smallPot = int(myRandom(0, 2));
        totalSupplies.largePot = int(myRandom(0, 1));
        totalSupplies.smallPan = int(myRandom(0, 2));
        totalSupplies.largePan = int(myRandom(0, 1));
        totalSupplies.soap = int(myRandom(0, 2));
        totalSupplies.clothes = int(myRandom(0, 5));
        totalSupplies.blankets = int(myRandom(0, 5));
        //
        getSupplies();
        toTalk = “fiveDime”;
        peopleSpeak(toTalk);
        attachRules(“fiveDime”);
        // Leaving Oklahoma
    } else if (frameCount == 5) {
        _root.movieClipLoader_mc.currentMC.exit_mc.unloadMovie();
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.informationMC.interface_mc._visible = false;
        _root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
            soundFX(“BUZZER”);
            if (frameCount<allKnowing.myFiles-1) {
                frameCount++;
                loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
            }
        };
        // Route 66 - 1
    } else if (frameCount == 6) {
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
```

```
		attachRules(“theRoad”);
		_root.informationMC.interface_mc._visible = true;
		_root.informationMC.interface_mc.gasGauge_mc._visible = true;
		_root.movieClipLoader_mc.currentMC.miles_txt.text = mileCounter;
		_root.movieClipLoader_mc.currentMC.gallons_txt.text = gasGallons;
		_root.movieClipLoader_mc.currentMC.mpg_txt.text = gasMileage;
		_root.movieClipLoader_mc.currentMC.gas_txt.text = calculateGas();
		_root.movieClipLoader_mc.currentMC.food_txt.text = foodWeight;
		_root.movieClipLoader_mc.currentMC.weight_txt.text = totalWeight;
		_root.movieClipLoader_mc.currentMC.cash_txt.text = “$”+totalCash;
		_root.movieClipLoader_mc.currentMC.cash_txt.maxChars = 4;
		mc1_mc._x = -10;
		mc2_mc._x = -10;
		mc3_mc._x = -10;
		_root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._xscale = 400;
		_root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._yscale = 400;
		_root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._xscale = 400;
		_root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._yscale = 400;
		_root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._xscale = 400;
		_root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._yscale = 400;
		_root.movieClipLoader_mc.currentMC.gauge_mc.needle_mc._rotation = 0;
		//
		_root.storyMC.theStory_mc.close_btn.onPress = function() {
			//_root.storyMC.unloadMovie();
			soundFX(“GO”);
		};
		//
		checkTotalMovement();
		// Welcome to Amarillo
	} else if (frameCount == 7) {
		alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
		_root.informationMC.interface_mc._visible = false;
		attachRules(“inTown”);
		_root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
		};
		_root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
			_root.movieClipLoader_mc.currentMC.unloadMovie();
			_root.createEmptyMovieClip(“gamePlay_mc”, 1000);
			_root.gamePlay_mc.loadMovie(“content/soil.swf”);
			_root.createEmptyMovieClip(“escape_mc”, 1001);
			_root.escape_mc.attachMovie(“advanceMC”, “advance_mc”, 1);
			_root.escape_mc._x = 700;
			_root.escape_mc._y = 0;
			soundFX(“BUZZER”);
			_root.escape_mc.advance_mc.onRelease = function() {
				_root.gamePlay_mc.unloadMovie();
				_root.escape_mc.unloadMovie();
				if (frameCount<allKnowing.myFiles-1) {
					frameCount++;
					loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
				}
			};
		};
		_root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
			myTownTitle = allKnowing.allCities[1].attributes.item;
			myTownInfo = allKnowing.allCities[1].attributes.info;
			_root.createEmptyMovieClip(“townInfo”, 1001);
			_root.townInfo.attachMovie(“townInfoMC”, “townInfo_mc”, 1);
			_root.townInfo.townInfo_mc._x = 200;
			_root.townInfo.townInfo_mc._y = 300;
			soundFX(“BUZZER”);
```

```
            alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 2, _root.townInfo.townInfo_mc);
            _root.townInfo.townInfo_mc.townName_txt.text = “The town of “+myTownTitle;
            _root.townInfo.townInfo_mc.townInfo_txt.text = myTownInfo;
            _root.townInfo.townInfo_mc.close_btn.onPress = function() {
                //trace(“Hello!”);
                soundFX(“POP”);
                _root.townInfo.unloadMovie();
            };
        };
        // Amarillo Town
    } else if (frameCount == 8) {
        _root.informationMC.interface_mc._visible = true;
        _root.talk_mc.talkScreen_mc._y = -1000;
        //gameClue = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        store1 = allKnowing.allCities[1].attributes.store1;
        store2 = allKnowing.allCities[1].attributes.store2;
        store3 = allKnowing.allCities[1].attributes.store3;
        store4 = allKnowing.allCities[1].attributes.store4;
        myClue = allKnowing.allCities[1].attributes.clue;
        myTown = allKnowing.allCities[1].attributes.item;
        //trace(“store1: “+store1+newline+”store2: “+store2+newline+”store3: “+store3+newline+”store4: “+store4);
        populateTown(store1, store2, store3, store4, myTown, myClue);
        // Route66 - 2
    } else if (frameCount == 9) {
        pushClues();
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.miles_txt.text = mileCounter;
        _root.movieClipLoader_mc.currentMC.gallons_txt.text = gasGallons;
        _root.movieClipLoader_mc.currentMC.mpg_txt.text = gasMileage;
        _root.movieClipLoader_mc.currentMC.gas_txt.text = calculateGas();
        _root.movieClipLoader_mc.currentMC.food_txt.text = foodWeight;
        _root.movieClipLoader_mc.currentMC.weight_txt.text = totalWeight;
        _root.movieClipLoader_mc.currentMC.cash_txt.text = totalCash;
        _root.movieClipLoader_mc.currentMC.cash_txt.maxChars = 4;
        mc1_mc._x = -10;
        mc2_mc._x = -10;
        mc3_mc._x = -10;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._yscale = 400;
        //
        puttputt = true;
        //
        checkTotalMovement();
        // Welcome to Pie Town
    } else if (frameCount == 10) {
        _root.informationMC.interface_mc._visible = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
        };
        _root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
            _root.movieClipLoader_mc.currentMC.unloadMovie();
            _root.createEmptyMovieClip(“gamePlay_mc”, 1000);
            _root.gamePlay_mc.loadMovie(“content/interaction2.swf”);
            _root.createEmptyMovieClip(“escape_mc”, 1001);
            _root.escape_mc.attachMovie(“advanceMC”, “advance_mc”, 1);
            _root.escape_mc._x = 700;
            _root.escape_mc._y = 0;
            soundFX(“BUZZER”);
            _root.escape_mc.advance_mc.onRelease = function() {
                _root.gamePlay_mc.unloadMovie();
                _root.escape_mc.unloadMovie();
                if (frameCount<allKnowing.myFiles-1) {
                    frameCount++;
                    loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
                }
            };
        };
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
            myTownTitle = allKnowing.allCities[2].attributes.item;
            myTownInfo = allKnowing.allCities[2].attributes.info;
            _root.createEmptyMovieClip(“townInfo”, 1001);
            _root.townInfo.attachMovie(“townInfoMC”, “townInfo_mc”, 1);
            _root.townInfo.townInfo_mc._x = 200;
            _root.townInfo.townInfo_mc._y = 300;
            alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 2, _root.townInfo.townInfo_mc);
            _root.townInfo.townInfo_mc.townName_txt.text = “The town of “+myTownTitle;
            _root.townInfo.townInfo_mc.townInfo_txt.text = myTownInfo;
            soundFX(“BUZZER”);
            _root.townInfo.townInfo_mc.close_btn.onPress = function() {
                soundFX(“POP”);
                _root.townInfo.unloadMovie();
            };
        };
        // Pie Town
    } else if (frameCount == 11) {
        _root.informationMC.interface_mc._visible = true;
        //gameClue = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.attachMovie(“modelT_mc”, “modelT_mc”, 25);
        _root.movieClipLoader_mc.currentMC.modelT_mc._x = 240;
        _root.movieClipLoader_mc.currentMC.modelT_mc._y = 190;
        store1 = allKnowing.allCities[2].attributes.store1;
        store2 = allKnowing.allCities[2].attributes.store2;
        store3 = allKnowing.allCities[2].attributes.store3;
        store4 = allKnowing.allCities[2].attributes.store4;
        myClue = allKnowing.allCities[2].attributes.clue;
        myTown = allKnowing.allCities[2].attributes.item;
        trace(“store1: “+store1+newline+”store2: “+store2+newline+”store3: “+store3+newline+”store4: “+store4);
        populateTown(store1, store2, store3, store4, myTown, myClue);
        // Route 66 - 3
    } else if (frameCount == 12) {
        pushClues();
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.miles_txt.text = mileCounter;
        _root.movieClipLoader_mc.currentMC.gallons_txt.text = gasGallons;
        _root.movieClipLoader_mc.currentMC.mpg_txt.text = gasMileage;
        _root.movieClipLoader_mc.currentMC.gas_txt.text = calculateGas();
        _root.movieClipLoader_mc.currentMC.food_txt.text = foodWeight;
        _root.movieClipLoader_mc.currentMC.weight_txt.text = totalWeight;
        _root.movieClipLoader_mc.currentMC.cash_txt.text = totalCash;
        _root.movieClipLoader_mc.currentMC.cash_txt.maxChars = 4;
        mc1_mc._x = -10;
        mc2_mc._x = -10;
        mc3_mc._x = -10;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._xscale = 400;
```

```
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._yscale = 400;
        //
        puttputt = true;
        //
        checkTotalMovement();
        // Welcome to Flagstaff
    } else if (frameCount == 13) {
        _root.informationMC.interface_mc._visible = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
        };
        _root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
            _root.movieClipLoader_mc.currentMC.unloadMovie();
            _root.createEmptyMovieClip(“gamePlay_mc”, 1000);
            _root.gamePlay_mc.loadMovie(“content/hoover.swf”);
            _root.createEmptyMovieClip(“escape_mc”, 1001);
            _root.escape_mc.attachMovie(“advanceMC”, “advance_mc”, 1);
            _root.escape_mc._x = 700;
            _root.escape_mc._y = 0;
            soundFX(“BUZZER”);
            _root.escape_mc.advance_mc.onRelease = function() {
                _root.gamePlay_mc.unloadMovie();
                _root.escape_mc.unloadMovie();
                if (frameCount<allKnowing.myFiles-1) {
                    frameCount++;
                    loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
                }
            };
        };
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
            myTownTitle = allKnowing.allCities[3].attributes.item;
            myTownInfo = allKnowing.allCities[3].attributes.info;
            _root.createEmptyMovieClip(“townInfo”, 1001);
            _root.townInfo.attachMovie(“townInfoMC”, “townInfo_mc”, 1);
            _root.townInfo.townInfo_mc._x = 200;
            _root.townInfo.townInfo_mc._y = 300;
            alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 2, _root.townInfo.townInfo_mc);
            _root.townInfo.townInfo_mc.townName_txt.text = “The town of “+myTownTitle;
            _root.townInfo.townInfo_mc.townInfo_txt.text = myTownInfo;
            soundFX(“BUZZER”);
            _root.townInfo.townInfo_mc.close_btn.onPress = function() {
                soundFX(“POP”);
                _root.townInfo.unloadMovie();
            };
        };
        // The Town of Flagstaff
    } else if (frameCount == 14) {
        _root.informationMC.interface_mc._visible = true;
        //gameClue = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.attachMovie(“modelT_mc”, “modelT_mc”, 25);
        _root.movieClipLoader_mc.currentMC.modelT_mc._x = 240;
        _root.movieClipLoader_mc.currentMC.modelT_mc._y = 190;
        store1 = allKnowing.allCities[3].attributes.store1;
        store2 = allKnowing.allCities[3].attributes.store2;
        store3 = allKnowing.allCities[3].attributes.store3;
        store4 = allKnowing.allCities[3].attributes.store4;
        myClue = allKnowing.allCities[3].attributes.clue;
        myTown = allKnowing.allCities[3].attributes.item;
        trace(“store1: “+store1+newline+”store2: “+store2+newline+”store3: “+store3+newline+”store4: “+store4);
        populateTown(store1, store2, store3, store4, myTown, myClue);
        // Route 66 -4
    } else if (frameCount == 15) {
        pushClues();
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.miles_txt.text = mileCounter;
        _root.movieClipLoader_mc.currentMC.gallons_txt.text = gasGallons;
        _root.movieClipLoader_mc.currentMC.mpg_txt.text = gasMileage;
        _root.movieClipLoader_mc.currentMC.gas_txt.text = calculateGas();
        _root.movieClipLoader_mc.currentMC.food_txt.text = foodWeight;
        _root.movieClipLoader_mc.currentMC.weight_txt.text = totalWeight;
        _root.movieClipLoader_mc.currentMC.cash_txt.text = totalCash;
        _root.movieClipLoader_mc.currentMC.cash_txt.maxChars = 4;
        mc1_mc._x = -10;
        mc2_mc._x = -10;
        mc3_mc._x = -10;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._yscale = 400;
        //
        puttputt = true;
        //
        checkTotalMovement();
        // Welcome to Needles
    } else if (frameCount == 16) {
        _root.informationMC.interface_mc._visible = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
        };
        _root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
            _root.movieClipLoader_mc.currentMC.unloadMovie();
            _root.createEmptyMovieClip(“gamePlay_mc”, 1000);
            _root.gamePlay_mc.loadMovie(“content/newDeal.swf”);
            _root.createEmptyMovieClip(“escape_mc”, 1001);
            _root.escape_mc.attachMovie(“advanceMC”, “advance_mc”, 1);
            _root.escape_mc._x = 700;
            _root.escape_mc._y = 0;
            soundFX(“BUZZER”);
            _root.escape_mc.advance_mc.onRelease = function() {
                _root.gamePlay_mc.unloadMovie();
                _root.escape_mc.unloadMovie();
                if (frameCount<allKnowing.myFiles-1) {
                    frameCount++;
                    loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
                }
            };
        };
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
            myTownTitle = allKnowing.allCities[4].attributes.item;
            myTownInfo = allKnowing.allCities[4].attributes.info;
            _root.createEmptyMovieClip(“townInfo”, 1001);
            _root.townInfo.attachMovie(“townInfoMC”, “townInfo_mc”, 1);
            _root.townInfo.townInfo_mc._x = 200;
            _root.townInfo.townInfo_mc._y = 300;
            alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 2, _root.townInfo.townInfo_mc);
            _root.townInfo.townInfo_mc.townName_txt.text = “The town of “+myTownTitle;
            _root.townInfo.townInfo_mc.townInfo_txt.text = myTownInfo;
            soundFX(“BUZZER”);
            _root.townInfo.townInfo_mc.close_btn.onPress = function() {
                soundFX(“POP”);
                _root.townInfo.unloadMovie();
            };
        };
        // The Town of Needles
    } else if (frameCount == 17) {
        _root.informationMC.interface_mc._visible = true;
        //gameClue = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.attachMovie(“modelT_mc”, “modelT_mc”, 25);
        _root.movieClipLoader_mc.currentMC.modelT_mc._x = 240;
        _root.movieClipLoader_mc.currentMC.modelT_mc._y = 190;
        store1 = allKnowing.allCities[4].attributes.store1;
        store2 = allKnowing.allCities[4].attributes.store2;
        store3 = allKnowing.allCities[4].attributes.store3;
        store4 = allKnowing.allCities[4].attributes.store4;
        myClue = allKnowing.allCities[4].attributes.clue;
        myTown = allKnowing.allCities[4].attributes.item;
        trace(“store1: “+store1+newline+”store2: “+store2+newline+”store3: “+store3+newline+”store4: “+store4);
        populateTown(store1, store2, store3, store4, myTown, myClue);
        // Route 66 - 5
    } else if (frameCount == 18) {
        pushClues();
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.miles_txt.text = mileCounter;
        _root.movieClipLoader_mc.currentMC.gallons_txt.text = gasGallons;
        _root.movieClipLoader_mc.currentMC.mpg_txt.text = gasMileage;
        _root.movieClipLoader_mc.currentMC.gas_txt.text = calculateGas();
        _root.movieClipLoader_mc.currentMC.food_txt.text = foodWeight;
        _root.movieClipLoader_mc.currentMC.weight_txt.text = totalWeight;
        _root.movieClipLoader_mc.currentMC.cash_txt.text = totalCash;
        _root.movieClipLoader_mc.currentMC.cash_txt.maxChars = 4;
        mc1_mc._x = -10;
        mc2_mc._x = -10;
        mc3_mc._x = -10;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._yscale = 400;
        //
        puttputt = true;
        //
        checkTotalMovement();
        // Welcome to San Bernardino
    } else if (frameCount == 19) {
        _root.informationMC.interface_mc._visible = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
        };
        _root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
            _root.movieClipLoader_mc.currentMC.unloadMovie();
            _root.createEmptyMovieClip(“gamePlay_mc”, 1000);
            _root.gamePlay_mc.loadMovie(“content/quiz.swf”);
            _root.createEmptyMovieClip(“escape_mc”, 1001);
            _root.escape_mc.attachMovie(“advanceMC”, “advance_mc”, 1);
            _root.escape_mc._x = 700;
            _root.escape_mc._y = 0;
            soundFX(“BUZZER”);
            _root.escape_mc.advance_mc.onRelease = function() {
                _root.gamePlay_mc.unloadMovie();
                _root.escape_mc.unloadMovie();
                if (frameCount<allKnowing.myFiles-1) {
                    frameCount++;
                    loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
                }
            };
        };
        _root.movieClipLoader_mc.currentMC.info_btn.onPress = function() {
            myTownTitle = allKnowing.allCities[5].attributes.item;
            myTownInfo = allKnowing.allCities[5].attributes.info;
            _root.createEmptyMovieClip(“townInfo”, 1001);
            _root.townInfo.attachMovie(“townInfoMC”, “townInfo_mc”, 1);
            _root.townInfo.townInfo_mc._x = 200;
            _root.townInfo.townInfo_mc._y = 300;
            alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 2, _root.townInfo.townInfo_mc);
            _root.townInfo.townInfo_mc.townName_txt.text = “The town of “+myTownTitle;
            _root.townInfo.townInfo_mc.townInfo_txt.text = myTownInfo;
            soundFX(“BUZZER”);
            _root.townInfo.townInfo_mc.close_btn.onPress = function() {
                soundFX(“POP”);
                _root.townInfo.unloadMovie();
            };
        };
        // The Town of San Bernardino
    } else if (frameCount == 20) {
        _root.informationMC.interface_mc._visible = true;
        //gameClue = false;
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.attachMovie(“modelT_mc”, “modelT_mc”, 25);
        _root.movieClipLoader_mc.currentMC.modelT_mc._x = 240;
        _root.movieClipLoader_mc.currentMC.modelT_mc._y = 190;
        store1 = allKnowing.allCities[5].attributes.store1;
        store2 = allKnowing.allCities[5].attributes.store2;
        store3 = allKnowing.allCities[5].attributes.store3;
        store4 = allKnowing.allCities[5].attributes.store4;
        myClue = allKnowing.allCities[5].attributes.clue;
        myTown = allKnowing.allCities[5].attributes.item;
        trace(“store1: “+store1+newline+”store2: “+store2+newline+”store3: “+store3+newline+”store4: “+store4);
        populateTown(store1, store2, store3, store4, myTown, myClue);
        // To the Migrant Camp
    } else if (frameCount == 21) {
        pushClues();
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.movieClipLoader_mc.currentMC.miles_txt.text = mileCounter;
        _root.movieClipLoader_mc.currentMC.gallons_txt.text = gasGallons;
        _root.movieClipLoader_mc.currentMC.mpg_txt.text = gasMileage;
        _root.movieClipLoader_mc.currentMC.gas_txt.text = calculateGas();
        _root.movieClipLoader_mc.currentMC.food_txt.text = foodWeight;
        _root.movieClipLoader_mc.currentMC.weight_txt.text = totalWeight;
        _root.movieClipLoader_mc.currentMC.cash_txt.text = totalCash;
        _root.movieClipLoader_mc.currentMC.cash_txt.maxChars = 4;
        mc1_mc._x = -10;
        mc2_mc._x = -10;
        mc3_mc._x = -10;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc._yscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._xscale = 400;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc._yscale = 400;
        //
        puttputt = true;
        //
        checkTotalMovement();
        // welcome to the Migrant Camp
    } else if (frameCount == 22) {
        _root.movieClipLoader_mc.currentMC.exit_mc.unloadMovie();
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.informationMC.interface_mc._visible = false;
        _root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
            soundFX(“BUZZER”);
            if (frameCount<allKnowing.myFiles-1) {
                frameCount++;
                loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
            }
        };
        //inside the Migrant Camp
    } else if (frameCount == 23) {
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        _root.informationMC.interface_mc._visible = true;
        tent1 = allKnowing.allCities[6].attributes.store1;
        tent2 = allKnowing.allCities[6].attributes.store2;
        tent3 = allKnowing.allCities[6].attributes.store3;
        tent4 = allKnowing.allCities[6].attributes.store4;
        myClue = allKnowing.allCities[6].attributes.clue;
        trace(“store1: “+store1+newline+”store2: “+store2+newline+”store3: “+store3+newline+”store4: “+store4);
        populateTents(tent1, tent2, tent3, tent4, myClue);
        moveScreen();
        //
        leftArrow = _root.createEmptyMovieClip(“leftArrow_mc”, 7654);
        leftArrow.attachMovie(“leftarrowMC”, “leftArrowMC”, 1);
        leftArrow.leftArrowMC._x = -1000;
        leftArrow.leftArrowMC._y = -1000;
        //
        rightArrow = _root.createEmptyMovieClip(“rightArrow_mc”, 7655);
        rightArrow.attachMovie(“rightarrowMC”, “rightArrowMC”, 1);
        rightArrow.rightArrowMC._x = -1000;
        rightArrow.rightArrowMC._y = -1000;
        //
        _root.movieClipLoader_mc.currentMC.attachMovie(“rearModelT”, “modelT_mc”, 9);
        _root.movieClipLoader_mc.currentMC.modelT_mc._x = 1050;
        _root.movieClipLoader_mc.currentMC.modelT_mc._y = 200;
        _root.movieClipLoader_mc.currentMC.modelT_mc._xscale = 66;
        _root.movieClipLoader_mc.currentMC.modelT_mc._yscale = 66;
        _root.movieClipLoader_mc.currentMC.modelT_mc.onRelease = function() {
            soundFX(“BUZZER”);
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                _root.talk_mc.talkScreen_mc._x = (Stage.width/2)+100;
                _root.talk_mc.talkScreen_mc._y = Stage.height/2;
                _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                _root.talk_mc.talkScreen_mc.item_txt.text = “Are you sure you want to leave?”;
                _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                    checkClues(4);
                    trace(“my clue value : “+checkClues);
                    if (clueValue == true) {
                        frameCount++;
                        _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.
                        SWFloc);
                        //finishGame(“end1MC”, allKnowing.allEnding[1].attributes.item);
                    }
                    if (clueValue == false) {
                        frameCount++;
                        //_root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.
                        SWFloc);
                        finishGame(“end1MC”, allKnowing.allEnding[1].attributes.item);
                    }
                    if (clueValue == undefined) {
                        frameCount++;
                        //_root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.
                        SWFloc);
                        finishGame(“end1MC”, allKnowing.allEnding[1].attributes.item);
                    }
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    soundFX(“POP”);
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                };
                _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    soundFX(“POP”);
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                };
            };
        };
        //Orange Grove
    } else if (frameCount == 24) {
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        toTalk = “orangeGrove”;
        peopleSpeak(toTalk);
        //_root.movieClipLoader_mc.currentMC.attachMovie(“exit_mc”, “exit_mc”, 2);
        //_root.movieClipLoader_mc.currentMC.exit_mc._x = 240;
        //_root.movieClipLoader_mc.currentMC.exit_mc._y = 200;
        //_root.movieClipLoader_mc.currentMC.exit_mc.onPress = function() {
        //frameCount++;
        //_root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        //};
        // The End / Scores
    } else if (frameCount == 25) {
        alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
        //
        _root.informationMC.interface_mc._visible = false;
        //
        markerScore = 0;
        clueScore = 0;
        //
        _root.movieClipLoader_mc.currentMC.family_txt.text = gamePlayerArray.toString();
        _root.movieClipLoader_mc.currentMC.point1_txt.text = int(gamePlayerArray.length)*200;
        point1 = int(gamePlayerArray.length)*200;
        //
        _root.movieClipLoader_mc.currentMC.flour_txt.text = foodSupplies.flour;
        _root.movieClipLoader_mc.currentMC.vegetable_txt.text = foodSupplies.vegetable;
        _root.movieClipLoader_mc.currentMC.potato_txt.text = foodSupplies.potato;
        _root.movieClipLoader_mc.currentMC.pork_txt.text = foodSupplies.pork;
        _root.movieClipLoader_mc.currentMC.coffee_txt.text = foodSupplies.coffee;
        _root.movieClipLoader_mc.currentMC.sugar_txt.text = foodSupplies.sugar;
        _root.movieClipLoader_mc.currentMC.point2_txt.text = (int(foodSupplies.flour)+int(foodSupplies.vegetable)+int(foodSupplies.
        potato)+int(foodSupplies.pork)+int(foodSupplies.coffee)+int(foodSupplies.sugar))*10;
        point2 = (int(foodSupplies.flour)+int(foodSupplies.vegetable)+int(foodSupplies.potato)+int(foodSupplies.pork)+int(foodSupplies.
        coffee)+int(foodSupplies.sugar))*10;
        //
        _root.movieClipLoader_mc.currentMC.route_txt.text = gameMarkerArray[0].toString();
```

```
                _root.movieClipLoader_mc.currentMC.santa_txt.text = gameMarkerArray[1].toString();
                _root.movieClipLoader_mc.currentMC.mexican_txt.text = gameMarkerArray[2].toString();
                _root.movieClipLoader_mc.currentMC.manifest_txt.text = gameMarkerArray[3].toString();
                _root.movieClipLoader_mc.currentMC.cali_txt.text = gameMarkerArray[4].toString();
                for (var m = 0; m<=(gameMarkerArray.length)-1; m++) {
                        if (gameMarkerArray[m] == “yes”) {
                                markerScore = markerScore+10;
                        } else {
                                null;
                        }
                }
                _root.movieClipLoader_mc.currentMC.point3_txt.text = int(markerScore);
                point3 = int(markerScore);
                //
                _root.movieClipLoader_mc.currentMC.letter_txt.text = gameCluesArray[0].toString();
                _root.movieClipLoader_mc.currentMC.orange_txt.text = gameCluesArray[2].toString();
                _root.movieClipLoader_mc.currentMC.card_txt.text = gameCluesArray[3].toString();
                _root.movieClipLoader_mc.currentMC.map_txt.text = gameCluesArray[1].toString();
                for (var c = 0; c<=(gameCluesArray.length)-1; c++) {
                        if (gameCluesArray[c] == true) {
                                clueScore = clueScore+50;
                        } else {
                                null;
                        }
                }
                _root.movieClipLoader_mc.currentMC.point4_txt.text = int(clueScore);
                point4 = int(clueScore);
                //
                gasScore = int(gasGallons)*10;
                cashScore = int(totalCash)*10;
                tripScore = int(gasScore)+int(cashScore)+int(happiness);
                _root.movieClipLoader_mc.currentMC.happy_txt.text = happiness;
                _root.movieClipLoader_mc.currentMC.gas_txt.text = gasScore;
                _root.movieClipLoader_mc.currentMC.cash_txt.text = cashScore;
                _root.movieClipLoader_mc.currentMC.triptotal_txt.text = tripScore;
                _root.movieClipLoader_mc.currentMC.travel_txt.text = dayCount;
                _root.movieClipLoader_mc.currentMC.penalty_txt.text = penaltyCount;
                _root.movieClipLoader_mc.currentMC.point5_txt.text = tripScore-dayCount-penaltyCount;
                point5 = tripScore-dayCount-penaltyCount;
                //
                totalScore = (int(point1)+int(point2)+int(point3)+int(point4)+int(point5));
                _root.movieClipLoader_mc.currentMC.total_txt.text = int(totalScore);
                //
                _root.movieClipLoader_mc.currentMC.feedback_mc.onRelease = function() {
                        getURL(“http://thesis.dkerrcreative.com/questions.html”, “_blank”);
                        soundFX(“BUZZER”);
                };
                _root.movieClipLoader_mc.currentMC.restart_mc.onRelease = function() {
                        _root.gotoAndPlay(1);
                        soundFX(“BUZZER”);
                };
        }
};
function alphaFade(easing, myBegin, myEnd, myTime, myMC) {
        var easeType = easing;
        var begin = myBegin;
        var end = myEnd;
        var time = myTime;
        var mc = myMC;
        myTween = new mx.transitions.Tween(mc, “_alpha”, easeType, begin, end, time, true);
}
```

Actionscript Code: Frame 2, Layer “XML Setup/Includes”

```
#include “includes/random.as”
#include “includes/moveCar.as”
#include “includes/foodPrices.as”
#include “includes/getSupplies.as”
#include “includes/checkTotalTrip.as”
#include “includes/populateTown.as”
#include “includes/populateTents.as”
#include “includes/testAmounts.as”
#include “includes/peopleSpeak.as”
#include “includes/endGame.as”

// Load First XML
showresults = function (success) {
    // trace(“results fired”);
    if (success == true) {
        parseXMLSetup();
    } else {
        // trace(“XML load failed”);
    }
};
// Handle XML
setup_xml = new XML();
setup_xml.ignoreWhite = true;
setup_xml.load(“XML/content.xml”);
setup_xml.onLoad = showresults;
//
parseXMLSetup = function () {
    allKnowing.myFiles = setup_xml.firstChild.childNodes[0].childNodes.length;
    allKnowing.allMCs = setup_xml.firstChild.childNodes[0].childNodes;
    allKnowing.allPrices = setup_xml.firstChild.childNodes[1].childNodes;
    allKnowing.allSupplies = setup_xml.firstChild.childNodes[2].childNodes;
    allKnowing.allCosts = setup_xml.firstChild.childNodes[3].childNodes;
    allKnowing.allCities = setup_xml.firstChild.childNodes[4].childNodes;
    allKnowing.allMarkers = setup_xml.firstChild.childNodes[5].childNodes;
    allKnowing.allVoices = setup_xml.firstChild.childNodes[6].childNodes;
    allKnowing.allEnding = setup_xml.firstChild.childNodes[7].childNodes;
    for (var j = 2; j<=(allKnowing.allPrices.length)-1; j++) {
        var myFood = allKnowing.allPrices[j].attributes.item;
        foodSupplies[myFood] = int(0);
    }
    for (var s = 0; s<=(allKnowing.allSupplies.length)-1; s++) {
        var myItem = allKnowing.allSupplies[s].attributes.id;
        totalSupplies[myItem] = int(0);
    }
    this.onEnterFrame = function() {
        loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        delete this.onEnterFrame;
    };
};
```

Actionscript Code: Include “FoodPrices.as”

```
getFood = function () {
    myPriceArray = new Array();
    for (var i = 2; i<=allKnowing.allPrices.length-1; i++) {
        var myMinPrice = allKnowing.allPrices[i].attributes.min;
        var myMaxPrice = allKnowing.allPrices[i].attributes.max;
        var myRandomNum = int(myRandom(int(myMinPrice), int(myMaxPrice)));
        myPriceArray.push((myRandomNum)*.01);
}
//
_root.movieClipLoader_mc.currentMC.flour_mc.onRollOver = function() {
        this.onEnterFrame = function() {
                foodItem = allKnowing.allPrices[2].attributes.item;
                foodPrice = myPriceArray[0];
                _root.desc_mc.itemDescription_mc._x = _xmouse;
                //this._x+400;
                _root.desc_mc.itemDescription_mc._y = _ymouse;
                //this._y+300;
                _root.desc_mc.itemDescription_mc.item_txt.text = foodItem;
                _root.desc_mc.itemDescription_mc.price_txt.text = “$”+foodPrice+” per pound”;
        };
};
_root.movieClipLoader_mc.currentMC.flour_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.flour_mc.onPress = function() {
        foodItem = allKnowing.allPrices[2].attributes.item;
        foodPrice = myPriceArray[0];
        buyFood(foodItem, foodPrice);
        soundFX(“BUZZER”);
        //_root.movieClipLoader_mc.currentMC.item_txt.text = foodItem;
        //_root.movieClipLoader_mc.currentMC.price_txt.text = “$”+foodPrice;
};
//
_root.movieClipLoader_mc.currentMC.potatoes_mc.onRollOver = function() {
        this.onEnterFrame = function() {
                foodItem = allKnowing.allPrices[3].attributes.item;
                foodPrice = myPriceArray[1];
                _root.desc_mc.itemDescription_mc._x = _xmouse;
                _root.desc_mc.itemDescription_mc._y = _ymouse;
                _root.desc_mc.itemDescription_mc.item_txt.text = foodItem;
                _root.desc_mc.itemDescription_mc.price_txt.text = “$”+foodPrice+” per pound”;
        };
};
_root.movieClipLoader_mc.currentMC.potatoes_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.potatoes_mc.onPress = function() {
        foodItem = allKnowing.allPrices[3].attributes.item;
        foodPrice = myPriceArray[1];
        buyFood(foodItem, foodPrice);
        soundFX(“BUZZER”);
        //_root.movieClipLoader_mc.currentMC.item_txt.text = foodItem;
        //_root.movieClipLoader_mc.currentMC.price_txt.text = “$”+foodPrice;
};
//
_root.movieClipLoader_mc.currentMC.vegetables_mc.onRollOver = function() {
        this.onEnterFrame = function() {
                foodItem = allKnowing.allPrices[4].attributes.item;
                foodPrice = myPriceArray[2];
                _root.desc_mc.itemDescription_mc._x = _xmouse;
                _root.desc_mc.itemDescription_mc._y = _ymouse;
                _root.desc_mc.itemDescription_mc.item_txt.text = foodItem;
                _root.desc_mc.itemDescription_mc.price_txt.text = “$”+foodPrice+” per pound”;
        };
};
_root.movieClipLoader_mc.currentMC.vegetables_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.vegetables_mc.onPress = function() {
        foodItem = allKnowing.allPrices[4].attributes.item;
        foodPrice = myPriceArray[2];
        buyFood(foodItem, foodPrice);
        soundFX(“BUZZER”);
        //_root.movieClipLoader_mc.currentMC.item_txt.text = foodItem;
        //_root.movieClipLoader_mc.currentMC.price_txt.text = “$”+foodPrice;
    };
    //
    _root.movieClipLoader_mc.currentMC.pork_mc.onRollOver = function() {
        this.onEnterFrame = function() {
            foodItem = allKnowing.allPrices[5].attributes.item;
            foodPrice = myPriceArray[3];
            _root.desc_mc.itemDescription_mc._x = _xmouse;
            _root.desc_mc.itemDescription_mc._y = _ymouse;
            _root.desc_mc.itemDescription_mc.item_txt.text = foodItem;
            _root.desc_mc.itemDescription_mc.price_txt.text = “$”+foodPrice+” per pound”;
        };
    };
    _root.movieClipLoader_mc.currentMC.pork_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.pork_mc.onPress = function() {
        foodItem = allKnowing.allPrices[5].attributes.item;
        foodPrice = myPriceArray[3];
        buyFood(foodItem, foodPrice);
        soundFX(“BUZZER”);
        //_root.movieClipLoader_mc.currentMC.item_txt.text = foodItem;
        //_root.movieClipLoader_mc.currentMC.price_txt.text = “$”+foodPrice;
    };
    //
    _root.movieClipLoader_mc.currentMC.coffee_mc.onRollOver = function() {
        this.onEnterFrame = function() {
            foodItem = allKnowing.allPrices[6].attributes.item;
            foodPrice = myPriceArray[4];
            _root.desc_mc.itemDescription_mc._x = _xmouse;
            _root.desc_mc.itemDescription_mc._y = _ymouse;
            _root.desc_mc.itemDescription_mc.item_txt.text = foodItem;
            _root.desc_mc.itemDescription_mc.price_txt.text = “$”+foodPrice+” per pound”;
        };
    };
    _root.movieClipLoader_mc.currentMC.coffee_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.coffee_mc.onPress = function() {
        foodItem = allKnowing.allPrices[6].attributes.item;
        foodPrice = myPriceArray[4];
        buyFood(foodItem, foodPrice);
        soundFX(“BUZZER”);
        //_root.movieClipLoader_mc.currentMC.item_txt.text = foodItem;
        //_root.movieClipLoader_mc.currentMC.price_txt.text = “$”+foodPrice;
    };
    //
    _root.movieClipLoader_mc.currentMC.sugar_mc.onRollOver = function() {
        this.onEnterFrame = function() {
            foodItem = allKnowing.allPrices[7].attributes.item;
            foodPrice = myPriceArray[5];
            _root.desc_mc.itemDescription_mc._x = _xmouse;
            _root.desc_mc.itemDescription_mc._y = _ymouse;
            _root.desc_mc.itemDescription_mc.item_txt.text = foodItem;
            _root.desc_mc.itemDescription_mc.price_txt.text = “$”+foodPrice+” per pound”;
        };
    };
    _root.movieClipLoader_mc.currentMC.sugar_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.sugar_mc.onPress = function() {
        foodItem = allKnowing.allPrices[7].attributes.item;
        foodPrice = myPriceArray[5];
        buyFood(foodItem, foodPrice);
        soundFX(“BUZZER”);
        //_root.movieClipLoader_mc.currentMC.item_txt.text = foodItem;
        //_root.movieClipLoader_mc.currentMC.price_txt.text = “$”+foodPrice;
    };
    //
};
addFood = function (selectedFood, action, difference) {
    trace(“hi food!”);
    var myFoodItem = selectedFood+”_txt”;
    if (action == “add”) {
        foodWeight = Number(foodWeight)+Number(difference);
        _root.informationMC.foodCounterMC[myFoodItem].text = foodSupplies[selectedFood];
        return foodWeight;
    }
    if (action == “minus”) {
        foodWeight = Number(foodWeight)-Number(difference);
        _root.informationMC.foodCounterMC[myFoodItem].text = foodSupplies[selectedFood];
        return foodWeight;
    }
    if (action == “delete”) {
        foodWeight = Number(foodWeight)-Number(difference);
        _root.informationMC.foodCounterMC[myFoodItem].text = 0;
        return foodWeight;
    }
};
buyFood = function (selectedFood, price) {
    _root.talk_mc.talkScreen_mc._y = -1000;
    _root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“buyLoadMC”, 1001);
    _root.movieClipLoader_mc.currentMC.buyLoadMC.attachMovie(“buyScreenMC”, “buyScreen_mc”, 5000);
    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._x = -350;
    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._y = -200;
    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.onEnterFrame = function() {
        if (_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._currentframe == 1) {
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.item_txt.text = selectedFood+” is “+price+” cents per pound.
            Would you like to buy the item?”;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.yes_mc.onPress = function() {
                _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(2);
                soundFX(“POP”);
            };
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.no_mc.onPress = function() {
                _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.unloadMovie();
                soundFX(“POP”);
            };
        }
        //
        if (_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._currentframe == 2) {
```

```
				_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.item_txt.text = "How many pounds would you like?";
				//_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = "";
				_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.maxChars = 2;
				_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.restrict = "0-9";
				_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.buy_mc.onPress = function() {
					soundFX("POP");
					var itemPrice = foodPrice;
					var itemAmount = _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text;
					var itemTotal = itemPrice*itemAmount;
					if (totalCash-itemTotal>0) {
						foodSupplies[selectedFood] = int(foodSupplies[selectedFood])+int(itemAmount);
						foodWeight = addFood(selectedFood, "add", itemAmount);
						totalWeight = int(foodWeight)+int(supplyWeight);
						_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = "";
						_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(3);
						storeMessage = "thank you!";
						checkCash(itemTotal, "minus");
						if (noFood == true) {
							noFood = false;
						}
					} else if (totalCash-itemTotal<0) {
						trace("you cant buy "+totalCash+" : "+itemTotal);
						_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = "";
						_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(4);
						storeMessage = "you don't have enough cash to buy these items";
						//checkCash(itemTotal, "minus");
					}
				};
			}
			//
			if (_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._currentframe == 3) {
				_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.item_txt.text = storeMessage;
				_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.close_mc.onPress = function() {
					_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.unloadMovie();
					soundFX("POP");
				};
			}
		};
};
```

Actionscript Code: Include "getSupplies.as"

```
getSupplies = function () {
	mySaleArray = new Array();
	myBuyArray = new Array();
	for (var i = 0; i<=allKnowing.allSupplies.length-1; i++) {
		var myMinPrice = allKnowing.allSupplies[i].attributes.min;
		var myMaxPrice = allKnowing.allSupplies[i].attributes.max;
		var myRandomSell = int(myRandom(int(myMinPrice), int(myMaxPrice)));
		mySaleArray.push((myRandomSell)*.01);
		var buyMinPrice = allKnowing.allSupplies[i].attributes.minPrice;
		var buyMaxPrice = allKnowing.allSupplies[i].attributes.maxPrice;
		var myRandomBuy = int(myRandom(int(buyMinPrice), int(buyMaxPrice)));
		myBuyArray.push((myRandomBuy)*.01);
	}
	_root.movieClipLoader_mc.currentMC.largePot_mc.onRollOver = function() {
		this.onEnterFrame = function() {
			supplyItem = allKnowing.allSupplies[0].attributes.id;
			myItem = allKnowing.allSupplies[0].attributes.item;
			mySell = mySaleArray[0];
			myBuy = myBuyArray[0];
```

```
				myWeight = allKnowing.allSupplies[0].attributes.weight;
				_root.desc_mc.itemDescription_mc._x = _xmouse;
				_root.desc_mc.itemDescription_mc._y = _ymouse;
				_root.desc_mc.itemDescription_mc.item_txt.text = myItem;
				_root.desc_mc.itemDescription_mc.price_txt.text = myBuy+" dollars per item";
			};
		};
		_root.movieClipLoader_mc.currentMC.largePot_mc.onRollOut = function() {
			delete this.onEnterFrame;
			_root.desc_mc.itemDescription_mc._y = -1000;
		};
		_root.movieClipLoader_mc.currentMC.largePot_mc.onPress = function() {
			supplyItem = allKnowing.allSupplies[0].attributes.id;
			myItem = allKnowing.allSupplies[0].attributes.item;
			mySell = mySaleArray[0];
			myBuy = myBuyArray[0];
			myWeight = allKnowing.allSupplies[0].attributes.weight;
			soundFX("BUZZER");
			getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
		};
		//
		_root.movieClipLoader_mc.currentMC.smallPot_mc.onRollOver = function() {
			this.onEnterFrame = function() {
				supplyItem = allKnowing.allSupplies[1].attributes.id;
				myItem = allKnowing.allSupplies[1].attributes.item;
				mySell = mySaleArray[1];
				myBuy = myBuyArray[1];
				myWeight = allKnowing.allSupplies[1].attributes.weight;
				_root.desc_mc.itemDescription_mc._x = _xmouse;
				_root.desc_mc.itemDescription_mc._y = _ymouse;
				_root.desc_mc.itemDescription_mc.item_txt.text = myItem;
				_root.desc_mc.itemDescription_mc.price_txt.text = myBuy+" dollars per item";
			};
		};
		_root.movieClipLoader_mc.currentMC.smallPot_mc.onRollOut = function() {
			delete this.onEnterFrame;
			_root.desc_mc.itemDescription_mc._y = -1000;
		};
		_root.movieClipLoader_mc.currentMC.smallPot_mc.onPress = function() {
			supplyItem = allKnowing.allSupplies[1].attributes.id;
			myItem = allKnowing.allSupplies[1].attributes.item;
			mySell = mySaleArray[1];
			myBuy = myBuyArray[1];
			myWeight = allKnowing.allSupplies[1].attributes.weight;
			soundFX("BUZZER");
			getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
		};
		//
		_root.movieClipLoader_mc.currentMC.largePan_mc.onRollOver = function() {
			this.onEnterFrame = function() {
				supplyItem = allKnowing.allSupplies[2].attributes.id;
				myItem = allKnowing.allSupplies[2].attributes.item;
				mySell = mySaleArray[2];
				myBuy = myBuyArray[2];
				myWeight = allKnowing.allSupplies[2].attributes.weight;
				_root.desc_mc.itemDescription_mc._x = _xmouse;
				_root.desc_mc.itemDescription_mc._y = _ymouse;
				_root.desc_mc.itemDescription_mc.item_txt.text = myItem;
				_root.desc_mc.itemDescription_mc.price_txt.text = myBuy+" dollars per item";
			};
		};
```

```
_root.movieClipLoader_mc.currentMC.largePan_mc.onRollOut = function() {
	delete this.onEnterFrame;
	_root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.largePan_mc.onPress = function() {
	supplyItem = allKnowing.allSupplies[2].attributes.id;
	myItem = allKnowing.allSupplies[2].attributes.item;
	mySell = mySaleArray[2];
	myBuy = myBuyArray[2];
	myWeight = allKnowing.allSupplies[2].attributes.weight;
	_root.movieClipLoader_mc.currentMC.weight_txt.text = myWeight;
	_root.movieClipLoader_mc.currentMC.item_txt.text = allKnowing.allSupplies[2].attributes.item;
	_root.movieClipLoader_mc.currentMC.sell_txt.text = “$”+mySell;
	_root.movieClipLoader_mc.currentMC.buy_txt.text = “$”+myBuy;
	soundFX(“BUZZER”);
	getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
};
//
_root.movieClipLoader_mc.currentMC.smallPan_mc.onRollOver = function() {
	this.onEnterFrame = function() {
		supplyItem = allKnowing.allSupplies[3].attributes.id;
		myItem = allKnowing.allSupplies[3].attributes.item;
		mySell = mySaleArray[3];
		myBuy = myBuyArray[3];
		myWeight = allKnowing.allSupplies[3].attributes.weight;
		_root.desc_mc.itemDescription_mc._x = _xmouse;
		_root.desc_mc.itemDescription_mc._y = _ymouse;
		_root.desc_mc.itemDescription_mc.item_txt.text = myItem;
		_root.desc_mc.itemDescription_mc.price_txt.text = myBuy+” dollars per item”;
	};
};
_root.movieClipLoader_mc.currentMC.smallPan_mc.onRollOut = function() {
	delete this.onEnterFrame;
	_root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.smallPan_mc.onPress = function() {
	supplyItem = allKnowing.allSupplies[3].attributes.id;
	mySell = mySaleArray[3];
	myBuy = myBuyArray[3];
	myWeight = allKnowing.allSupplies[3].attributes.weight;
	_root.movieClipLoader_mc.currentMC.weight_txt.text = myWeight;
	_root.movieClipLoader_mc.currentMC.item_txt.text = allKnowing.allSupplies[3].attributes.item;
	_root.movieClipLoader_mc.currentMC.sell_txt.text = “$”+mySell;
	_root.movieClipLoader_mc.currentMC.buy_txt.text = “$”+myBuy;
	soundFX(“BUZZER”);
	getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
};
//
_root.movieClipLoader_mc.currentMC.cleaning_mc.onRollOver = function() {
	this.onEnterFrame = function() {
		supplyItem = allKnowing.allSupplies[4].attributes.id;
		myItem = allKnowing.allSupplies[4].attributes.item;
		mySell = mySaleArray[4];
		myBuy = myBuyArray[4];
		myWeight = allKnowing.allSupplies[4].attributes.weight;
		_root.desc_mc.itemDescription_mc._x = _xmouse;
		_root.desc_mc.itemDescription_mc._y = _ymouse;
		_root.desc_mc.itemDescription_mc.item_txt.text = myItem;
		_root.desc_mc.itemDescription_mc.price_txt.text = myBuy+” dollars per item”;
	};
};
```

```
_root.movieClipLoader_mc.currentMC.cleaning_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.cleaning_mc.onPress = function() {
        supplyItem = allKnowing.allSupplies[4].attributes.id;
        myItem = allKnowing.allSupplies[4].attributes.item;
        mySell = mySaleArray[4];
        myBuy = myBuyArray[4];
        myWeight = allKnowing.allSupplies[4].attributes.weight;
        _root.movieClipLoader_mc.currentMC.weight_txt.text = myWeight;
        _root.movieClipLoader_mc.currentMC.item_txt.text = allKnowing.allSupplies[4].attributes.item;
        _root.movieClipLoader_mc.currentMC.sell_txt.text = “$”+mySell;
        _root.movieClipLoader_mc.currentMC.buy_txt.text = “$”+myBuy;
        soundFX(“BUZZER”);
        getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
};
//
_root.movieClipLoader_mc.currentMC.clothes_mc.onRollOver = function() {
        this.onEnterFrame = function() {
                supplyItem = allKnowing.allSupplies[5].attributes.id;
                myItem = allKnowing.allSupplies[5].attributes.item;
                mySell = mySaleArray[5];
                myBuy = myBuyArray[5];
                myWeight = allKnowing.allSupplies[5].attributes.weight;
                _root.desc_mc.itemDescription_mc._x = _xmouse;
                _root.desc_mc.itemDescription_mc._y = _ymouse;
                _root.desc_mc.itemDescription_mc.item_txt.text = myItem;
                _root.desc_mc.itemDescription_mc.price_txt.text = myBuy+” dollars per item”;
        };
};
_root.movieClipLoader_mc.currentMC.clothes_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.clothes_mc.onPress = function() {
        supplyItem = allKnowing.allSupplies[5].attributes.id;
        myItem = allKnowing.allSupplies[5].attributes.item;
        mySell = mySaleArray[5];
        myBuy = myBuyArray[5];
        myWeight = allKnowing.allSupplies[5].attributes.weight;
        _root.movieClipLoader_mc.currentMC.weight_txt.text = myWeight;
        _root.movieClipLoader_mc.currentMC.item_txt.text = allKnowing.allSupplies[5].attributes.item;
        _root.movieClipLoader_mc.currentMC.sell_txt.text = “$”+mySell;
        _root.movieClipLoader_mc.currentMC.buy_txt.text = “$”+myBuy;
        soundFX(“BUZZER”);
        getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
};
//
_root.movieClipLoader_mc.currentMC.blankets_mc.onRollOver = function() {
        this.onEnterFrame = function() {
                supplyItem = allKnowing.allSupplies[6].attributes.id;
                myItem = allKnowing.allSupplies[6].attributes.item;
                mySell = mySaleArray[6];
                myBuy = myBuyArray[6];
                myWeight = allKnowing.allSupplies[6].attributes.weight;
                _root.desc_mc.itemDescription_mc._x = _xmouse;
                _root.desc_mc.itemDescription_mc._y = _ymouse;
                _root.desc_mc.itemDescription_mc.item_txt.text = myItem;
                _root.desc_mc.itemDescription_mc.price_txt.text = myBuy+” dollars per item”;
        };
```

```
};
_root.movieClipLoader_mc.currentMC.blankets_mc.onRollOut = function() {
    delete this.onEnterFrame;
    _root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.blankets_mc.onPress = function() {
    supplyItem = allKnowing.allSupplies[6].attributes.id;
    myItem = allKnowing.allSupplies[6].attributes.item;
    mySell = mySaleArray[6];
    myBuy = myBuyArray[6];
    myWeight = allKnowing.allSupplies[6].attributes.weight;
    _root.movieClipLoader_mc.currentMC.weight_txt.text = myWeight;
    _root.movieClipLoader_mc.currentMC.item_txt.text = allKnowing.allSupplies[6].attributes.item;
    _root.movieClipLoader_mc.currentMC.sell_txt.text = “$”+mySell;
    _root.movieClipLoader_mc.currentMC.buy_txt.text = “$”+myBuy;
    soundFX(“BUZZER”);
    getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
};
//
_root.movieClipLoader_mc.currentMC.largeWater_mc.onRollOver = function() {
    this.onEnterFrame = function() {
        supplyItem = allKnowing.allSupplies[7].attributes.id;
        myItem = allKnowing.allSupplies[7].attributes.item;
        mySell = mySaleArray[7];
        myBuy = myBuyArray[7];
        myWeight = allKnowing.allSupplies[7].attributes.weight;
        _root.desc_mc.itemDescription_mc._x = _xmouse;
        _root.desc_mc.itemDescription_mc._y = _ymouse;
        _root.desc_mc.itemDescription_mc.item_txt.text = myItem;
        _root.desc_mc.itemDescription_mc.price_txt.text = myBuy+” dollars per item”;
    };
};
_root.movieClipLoader_mc.currentMC.largeWater_mc.onRollOut = function() {
    delete this.onEnterFrame;
    _root.desc_mc.itemDescription_mc._y = -1000;
};
_root.movieClipLoader_mc.currentMC.largeWater_mc.onPress = function() {
    supplyItem = allKnowing.allSupplies[7].attributes.id;
    myItem = allKnowing.allSupplies[7].attributes.item;
    mySell = mySaleArray[7];
    myBuy = myBuyArray[7];
    myWeight = allKnowing.allSupplies[7].attributes.weight;
    _root.movieClipLoader_mc.currentMC.weight_txt.text = myWeight;
    _root.movieClipLoader_mc.currentMC.item_txt.text = allKnowing.allSupplies[7].attributes.item;
    _root.movieClipLoader_mc.currentMC.sell_txt.text = “$”+mySell;
    _root.movieClipLoader_mc.currentMC.buy_txt.text = “$”+myBuy;
    soundFX(“BUZZER”);
    getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
};
//
_root.movieClipLoader_mc.currentMC.smallWater_mc.onRollOver = function() {
    this.onEnterFrame = function() {
        supplyItem = allKnowing.allSupplies[8].attributes.id;
        myItem = allKnowing.allSupplies[8].attributes.item;
        mySell = mySaleArray[8];
        myBuy = myBuyArray[8];
        myWeight = allKnowing.allSupplies[8].attributes.weight;
        _root.desc_mc.itemDescription_mc._x = _xmouse;
        _root.desc_mc.itemDescription_mc._y = _ymouse;
        _root.desc_mc.itemDescription_mc.item_txt.text = myItem;
        _root.desc_mc.itemDescription_mc.price_txt.text = myBuy+” dollars per item”;
        };
    };
    _root.movieClipLoader_mc.currentMC.smallWater_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.desc_mc.itemDescription_mc._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.smallWater_mc.onPress = function() {
        supplyItem = allKnowing.allSupplies[8].attributes.id;
        myItem = allKnowing.allSupplies[8].attributes.item;
        mySell = mySaleArray[8];
        myBuy = myBuyArray[8];
        myWeight = allKnowing.allSupplies[8].attributes.weight;
        getMySupplies(myItem, myBuy, mySell, itemWeight, supplyItem);
    };
    _root.movieClipLoader_mc.currentMC.buy_mc.onPress = function() {
        var itemPrice = myBuy;
        var itemAmount = _root.movieClipLoader_mc.currentMC.amount_txt.text;
        var itemWeight = myWeight*itemAmount;
        var itemTotal = itemPrice*itemAmount;
        supplyWeight = addSupplies(supplyItem, “add”, itemWeight);
        totalWeight = int(foodWeight)+int(supplyWeight);
        _root.movieClipLoader_mc.currentMC.cash_txt.text = “$”+checkCash(itemTotal, “minus”);
        _root.movieClipLoader_mc.currentMC.supply_txt.text = supplyWeight;
        _root.movieClipLoader_mc.currentMC.total_txt.text = totalWeight;
        _root.movieClipLoader_mc.currentMC.item_txt.text = “”;
        _root.movieClipLoader_mc.currentMC.amount_txt.text = “”;
    };
    _root.movieClipLoader_mc.currentMC.sell_mc.onPress = function() {
        var itemPrice = mySell;
        var itemAmount = _root.movieClipLoader_mc.currentMC.amount_txt.text;
        var itemWeight = myWeight*itemAmount;
        var itemTotal = itemPrice*itemAmount;
        supplyWeight = addSupplies(supplyItem, “minus”, itemWeight);
        totalWeight = int(foodWeight)+int(supplyWeight);
        _root.movieClipLoader_mc.currentMC.cash_txt.text = “$”+checkCash(itemTotal, “add”);
        _root.movieClipLoader_mc.currentMC.supply_txt.text = supplyWeight;
        _root.movieClipLoader_mc.currentMC.total_txt.text = totalWeight;
        _root.movieClipLoader_mc.currentMC.item_txt.text = “”;
        _root.movieClipLoader_mc.currentMC.amount_txt.text = “”;
    };
};
//add
addSupplies = function (supplyItem, action, difference) {
    var mySupply = supplyItem+”_txt”;
    if (action == “add”) {
        supplyWeight = int(supplyWeight)+int(difference);
        totalSupplies[supplyItem] = int(totalSupplies[supplyItem])+int(difference);
        _root.informationMC.supplyCounterMC[mySupply].text = totalSupplies[supplyItem];
    }
    if (action == “minus”) {
        supplyWeight = int(supplyWeight)-int(difference);
        totalSupplies[supplyItem] = int(totalSupplies[supplyItem])-int(difference);
        _root.informationMC.supplyCounterMC[mySupply].text = totalSupplies[supplyItem];
    }
    return supplyWeight;
};
getMySupplies = function (selectedItems, buyPrice, sellPrice, itemWeight, supplyItem) {
    _root.talk_mc.talkScreen_mc._y = -1000;
    _root.movieClipLoader_mc.currentMC.createEmptyMovieClip(“buyLoadMC”, 1001);
    _root.movieClipLoader_mc.currentMC.buyLoadMC.attachMovie(“buyScreen2MC”, “buyScreen_mc”, 5000);
    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._x = 125;
    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._y = -225;
    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.onEnterFrame = function() {
        if (_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._currentframe == 1) {
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.item_txt.text = selectedItems+” is $”+buyPrice+” per item to buy
and $”+sellPrice+” per item to sell?”;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.buy_mc.onPress = function() {
                _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(2);
                soundFX(“POP”);
            };
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.sell_mc.onPress = function() {
                _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(3);
                soundFX(“POP”);
            };
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.close_mc.onPress = function() {
                _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.unloadMovie();
                soundFX(“POP”);
            };
        }
        //
        if (_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._currentframe == 2) {
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.item_txt.text = “How many items would you like to buy?”;
            //_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = “”;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.maxChars = 2;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.restrict = “0-9”;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.buy_mc.onPress = function() {
                soundFX(“POP”);
                var itemPrice = buyPrice;
                var itemAmount = _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text;
                var itemWeight = myWeight*itemAmount;
                var itemTotal = itemPrice*itemAmount;
                if (totalCash-itemTotal>0) {
                    trace(“you can buy “+totalCash+” : “+itemTotal);
                    supplyWeight = addSupplies(supplyItem, “add”, itemAmount);
                    totalWeight = int(supplyItem)+int(supplyWeight);
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = “”;
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(4);
                    storeMessage = “thank you!”;
                    checkCash(itemTotal, “minus”);
                } else if (totalCash-itemTotal<0) {
                    trace(“you cant buy “+totalCash+” : “+itemTotal);
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = “”;
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(4);
                    storeMessage = “you don’t have enough cash to buy these items”;
                    //checkCash(itemTotal, “minus”);
                }
            };
        }
        //
        if (_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._currentframe == 3) {
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.item_txt.text = “How many items would you like to sell?”;
            //_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = “”;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.maxChars = 2;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.restrict = “0-9”;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.buy_mc.onPress = function() {
                soundFX(“POP”);
                var itemPrice = sellPrice;
                var itemAmount = _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text;
                var itemWeight = myWeight*itemAmount;
                var itemTotal = itemPrice*itemAmount;
                trace(supplyItem+” : “+totalSupplies[supplyItem]+” : “+itemAmount);
                if (itemAmount<=totalSupplies[supplyItem]) {
                    supplyWeight = addSupplies(supplyItem, “minus”, itemAmount);
                    totalWeight = int(supplyItem)+int(supplyWeight);
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = “”;
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(4);
                    storeMessage = “thank you!”;
                    checkCash(itemTotal, “add”);
                } else if (itemAmount>totalSupplies[supplyItem]) {
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.pounds_txt.text = “”;
                    _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.gotoAndStop(4);
                    storeMessage = “you don’t have enough items to sell!”;
                }
            };
        }
        if (_root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc._currentframe == 4) {
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.item_txt.text = storeMessage;
            _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.close_mc.onPress = function() {
                soundFX(“POP”);
                _root.movieClipLoader_mc.currentMC.buyLoadMC.buyScreen_mc.unloadMovie();
            };
        }
    };
};
```

Actionscript Code: Include “PeopleSpeak.as”

```
peopleSpeak = function (myPerson) {
    var myLines = 1;
    var gmaeClue = false;
    // Grocery
    if (myPerson == allKnowing.allVoices[0].attributes.item) {
        _root.movieClipLoader_mc.currentMC.groceryPerson_mc.onRollOver = function() {
            this.onEnterFrame = function() {
                trace(“where’s the dialog? “+_root.speak_mc.speakMC._y);
                _root.speak_mc.speakMC._x = _xmouse;
                _root.speak_mc.speakMC._y = _ymouse-10;
            };
        };
        _root.movieClipLoader_mc.currentMC.groceryPerson_mc.onRollOut = function() {
            delete this.onEnterFrame;
            _root.speak_mc.speakMC._y = -1000;
        };
        _root.movieClipLoader_mc.currentMC.groceryPerson_mc.onPress = function() {
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                _root.talk_mc.talkScreen_mc._x = 50;
                _root.talk_mc.talkScreen_mc._y = 100;
                var myWords = “speak”+myLines;
                _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[0].attributes[myWords];
                var myLength = allKnowing.allVoices[0].attributes.dialog;
                if (myLines<myLength) {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                } else {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                    delete this.onEnterFrame;
                }
                _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                    soundFX(“POP”);
                    myLines++;
                };
                _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                    soundFX(“POP”);
                    myLines = 1;
                };
                _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    soundFX(“POP”);
                    myLines = 1;
                };
            };
        };
    }
    // Supplies
    if (myPerson == allKnowing.allVoices[1].attributes.item) {
        _root.movieClipLoader_mc.currentMC.fiveDimePerson_mc.onRollOver = function() {
            this.onEnterFrame = function() {
                _root.speak_mc.speakMC._x = _xmouse;
                _root.speak_mc.speakMC._y = _ymouse-10;
            };
        };
        _root.movieClipLoader_mc.currentMC.fiveDimePerson_mc.onRollOut = function() {
            delete this.onEnterFrame;
            _root.speak_mc.speakMC._y = -1000;
        };
        _root.movieClipLoader_mc.currentMC.fiveDimePerson_mc.onPress = function() {
            _root.talk_mc.talkScreen_mc._x = 600;
            _root.talk_mc.talkScreen_mc._y = 100;
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                var myWords = “speak”+myLines;
                _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[1].attributes[myWords];
                var myLength = allKnowing.allVoices[1].attributes.dialog;
                if (myLines<myLength) {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                } else {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                    delete this.onEnterFrame;
                }
                _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                    soundFX(“POP”);
                    myLines++;
                };
                _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                    soundFX(“POP”);
                    myLines = 1;
                };
                _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    soundFX(“POP”);
                    myLines = 1;
                };
            };
        };
    }
    // Pie Town Community Center
    if (myPerson == allKnowing.allVoices[2].attributes.item) {
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onRollOver = function() {
            this.onEnterFrame = function() {
                _root.speak_mc.speakMC._x = _xmouse;
                _root.speak_mc.speakMC._y = _ymouse-10;
            };
        };
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onRollOut = function() {
            delete this.onEnterFrame;
            _root.speak_mc.speakMC._y = -1000;
        };
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onPress = function() {
            _root.talk_mc.talkScreen_mc._x = 100;
            _root.talk_mc.talkScreen_mc._y = 100;
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                var myWords = “speak”+myLines;
                _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[2].attributes[myWords];
                var myLength = allKnowing.allVoices[2].attributes.dialog;
                if (myLines<myLength) {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                } else {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                    delete this.onEnterFrame;
                }
                _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                    _root.informationMC.interface_mc.mapInfo_mc._visible = true;
                    soundFX(“POP”);
                    myLines++;
                };
                _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                    soundFX(“POP”);
                    gameClue = false;
                    myLines = 1;
                };
                _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    soundFX(“POP”);
                    gameClue = true;
                    myLines = 1;
                };
            };
        };
    }
    // Needles Community Center
    if (myPerson == allKnowing.allVoices[3].attributes.item) {
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onRollOver = function() {
            this.onEnterFrame = function() {
                _root.speak_mc.speakMC._x = _xmouse;
                _root.speak_mc.speakMC._y = _ymouse-10;
            };
        };
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onRollOut = function() {
            delete this.onEnterFrame;
            _root.speak_mc.speakMC._y = -1000;
        };
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onPress = function() {
            _root.talk_mc.talkScreen_mc._x = 100;
            _root.talk_mc.talkScreen_mc._y = 100;
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                var myWords = “speak”+myLines;
                _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[3].attributes[myWords];
                var myLength = allKnowing.allVoices[3].attributes.dialog;
                if (myLines<myLength) {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                } else {
```

```
				_root.talk_mc.talkScreen_mc.gotoAndStop(1);
				delete this.onEnterFrame;
			}
			if (myLines == 3) {
				attachClues(“registrationMC”);
			}
			_root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
				soundFX(“POP”);
				myLines++;
			};
			_root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
				_root.talk_mc.talkScreen_mc._y = -1000;
				delete _root.talk_mc.talkScreen_mc.onEnterFrame;
				soundFX(“POP”);
				gameClue = false;
				myLines = 1;
			};
			_root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
				_root.talk_mc.talkScreen_mc._y = -1000;
				soundFX(“POP”);
				gameClue = true;
				myLines = 1;
				attachClues(“unload”);
			};
		};
	};
}
// Amarillo Soda Fountain
if (myPerson == allKnowing.allVoices[4].attributes.item) {
	_root.movieClipLoader_mc.currentMC.sodaPerson_mc.onRollOver = function() {
		this.onEnterFrame = function() {
			trace(“where’s the dialog? “+_root.speak_mc.speakMC._y);
			_root.speak_mc.speakMC._x = _xmouse;
			_root.speak_mc.speakMC._y = _ymouse-10;
		};
	};
	_root.movieClipLoader_mc.currentMC.sodaPerson_mc.onRollOut = function() {
		delete this.onEnterFrame;
		_root.speak_mc.speakMC._y = -1000;
	};
	_root.movieClipLoader_mc.currentMC.sodaPerson_mc.onPress = function() {
		_root.talk_mc.talkScreen_mc._x = 100;
		_root.talk_mc.talkScreen_mc._y = 100;
		_root.talk_mc.talkScreen_mc.onEnterFrame = function() {
			var myWords = “speak”+myLines;
			_root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[4].attributes[myWords];
			var myLength = allKnowing.allVoices[4].attributes.dialog;
			if (myLines<myLength) {
				_root.talk_mc.talkScreen_mc.gotoAndStop(2);
			} else {
				_root.talk_mc.talkScreen_mc.gotoAndStop(1);
				delete this.onEnterFrame;
			}
			if (myLines == 3) {
				attachClues(“letterMC”);
			}
			_root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
				soundFX(“POP”);
				myLines++;
			};
			_root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
```

```
                _root.talk_mc.talkScreen_mc._y = -1000;
                delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                soundFX(“POP”);
                gameClue = false;
                myLines = 1;
            };
            _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                _root.talk_mc.talkScreen_mc._y = -1000;
                soundFX(“POP”);
                gameClue = true;
                myLines = 1;
                attachClues(“unload”);
            };
        };
    };
}
// Flagstaff Soup Kitchen
if (myPerson == allKnowing.allVoices[5].attributes.item) {
    _root.movieClipLoader_mc.currentMC.soupPerson_mc.onRollOver = function() {
        this.onEnterFrame = function() {
            trace(“where’s the dialog? “+_root.speak_mc.speakMC._y);
            _root.speak_mc.speakMC._x = _xmouse;
            _root.speak_mc.speakMC._y = _ymouse-10;
        };
    };
    _root.movieClipLoader_mc.currentMC.soupPerson_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.speak_mc.speakMC._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.soupPerson_mc.onPress = function() {
        _root.talk_mc.talkScreen_mc._x = 200;
        _root.talk_mc.talkScreen_mc._y = 100;
        _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
            var myWords = “speak”+myLines;
            _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[5].attributes[myWords];
            var myLength = allKnowing.allVoices[5].attributes.dialog;
            if (myLines<myLength) {
                _root.talk_mc.talkScreen_mc.gotoAndStop(2);
            } else {
                _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                delete this.onEnterFrame;
            }
            if (myLines == 3) {
                attachClues(“handbillMC”);
            }
            _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                soundFX(“POP”);
                myLines++;
            };
            _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                _root.talk_mc.talkScreen_mc._y = -1000;
                delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                soundFX(“POP”);
                gameClue = false;
                myLines = 1;
            };
            _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                howMuchFood = int(gamePlayerArray.length)*.25;
                //trace (“pounds added : “+howMuchFood)
                foodSupplies.flour = foodSupplies.flour+howMuchFood;
                foodSupplies.potato = foodSupplies.potato+howMuchFood;
```

```
                    foodSupplies.vegetable = foodSupplies.flour+howMuchFood;
                    foodSupplies.pork = foodSupplies.potato+howMuchFood;
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    soundFX(“POP”);
                    gameClue = true;
                    myLines = 1;
                    attachClues(“unload”);
                };
            };
        };
    }
    // San Bernardino Community Center
    if (myPerson == allKnowing.allVoices[6].attributes.item) {
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onRollOver = function() {
            this.onEnterFrame = function() {
                _root.speak_mc.speakMC._x = _xmouse;
                _root.speak_mc.speakMC._y = _ymouse-10;
            };
        };
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onRollOut = function() {
            delete this.onEnterFrame;
            _root.speak_mc.speakMC._y = -1000;
        };
        _root.movieClipLoader_mc.currentMC.communityPerson_mc.onPress = function() {
            _root.talk_mc.talkScreen_mc._x = 100;
            _root.talk_mc.talkScreen_mc._y = 100;
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                checkClues(3);
                //trace(“my clues are: “+clueValue);
                var myWords = “speak”+myLines;
                _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[6].attributes[myWords];
                var myLength = allKnowing.allVoices[6].attributes.dialog;
                if (clueValue == true) {
                    //trace(“Truth!”);
                    if (myLines<myLength) {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                    } else {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        gameClue = true;
                        pushClues();
                        delete this.onEnterFrame;
                    }
                    if (myLines == 2) {
                        attachClues(“workerCardMC”);
                    }
                    _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                        soundFX(“POP”);
                        myLines++;
                    };
                    _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        soundFX(“POP”);
                        myLines = 1;
                    };
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        soundFX(“POP”);
                        myLines = 1;
                        attachClues(“unload”);
                    };
                } else if (clueValue == false) {
                    //trace(“BOGUS!”);
                    if (myLines<myLength) {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                    } else {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        _root.talk_mc.talkScreen_mc.item_txt.text = “I’m sorry, you don’t have it. Go to the migrant camp outside of town
                        to look for work”;
                        delete this.onEnterFrame;
                    }
                    _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        //delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 2;
                    };
                    _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 1;
                    };
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        soundFX(“POP”);
                        gameClue = false;
                        pushClues();
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        myLines = 1;
                    };
                }
            };
        };
    }
    // Migrant Tent 1
    if (myPerson == allKnowing.allVoices[7].attributes.item) {
        _root.movieClipLoader_mc.currentMC.migrantFamily1_mc.onRollOver = function() {
            this.onEnterFrame = function() {
                trace(“where’s the dialog? “ + _root.speak_mc.speakMC._y);
                _root.speak_mc.speakMC._x = _xmouse
                _root.speak_mc.speakMC._y = _ymouse-10
            };
        };
        _root.movieClipLoader_mc.currentMC.migrantFamily1_mc.onRollOut = function() {
            delete this.onEnterFrame;
            _root.speak_mc.speakMC._y = -1000;
        };
        _root.movieClipLoader_mc.currentMC.migrantFamily1_mc.onPress = function() {
            _root.talk_mc.talkScreen_mc._x = 500;
            _root.talk_mc.talkScreen_mc._y = 250;
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                var myWords = “speak”+myLines;
                _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[7].attributes[myWords];
                var myLength = allKnowing.allVoices[7].attributes.dialog;
                if (myLines<myLength) {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                } else {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                    delete this.onEnterFrame;
                }
                _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                    soundFX(“POP”);
                    myLines++;
                };
                _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                    soundFX(“POP”);
                    myLines = 1;
                };
                _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    soundFX(“POP”);
                    myLines = 1;
                };
            };
        };
    }
    // Migrant Tent 2
    if (myPerson == allKnowing.allVoices[8].attributes.item) {
        _root.movieClipLoader_mc.currentMC.migrantFamily2_mc.onRollOver = function() {
            this.onEnterFrame = function() {
                trace(“where’s the dialog? “ + _root.speak_mc.speakMC._y);
                _root.speak_mc.speakMC._x = _xmouse
                _root.speak_mc.speakMC._y = _ymouse-10
            };
        };
        _root.movieClipLoader_mc.currentMC.migrantFamily2_mc.onRollOut = function() {
            delete this.onEnterFrame;
            _root.speak_mc.speakMC._y = -1000;
        };
        _root.movieClipLoader_mc.currentMC.migrantFamily2_mc.onPress = function() {
            _root.talk_mc.talkScreen_mc._x = 500;
            _root.talk_mc.talkScreen_mc._y = 250;
            _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
                var myWords = “speak”+myLines;
                _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[8].attributes[myWords];
                var myLength = allKnowing.allVoices[8].attributes.dialog;
                if (myLines<myLength) {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                } else {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                    delete this.onEnterFrame;
                }
                _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                    soundFX(“POP”);
                    myLines++;
                };
                _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                    soundFX(“POP”);
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                    myLines = 1;
                };
                _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                    soundFX(“POP”);
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    myLines = 1;
                };
            };
        };
    }
    // Migrant Tent 3
```

```
if (myPerson == allKnowing.allVoices[9].attributes.item) {
	_root.movieClipLoader_mc.currentMC.migrantFamily3_mc.onRollOver = function() {
		this.onEnterFrame = function() {
			trace("where's the dialog? " + _root.speak_mc.speakMC._y);
			_root.speak_mc.speakMC._x = _xmouse
			_root.speak_mc.speakMC._y = _ymouse-10
		};
	};
	_root.movieClipLoader_mc.currentMC.migrantFamily3_mc.onRollOut = function() {
		delete this.onEnterFrame;
		_root.speak_mc.speakMC._y = -1000;
	};
	_root.movieClipLoader_mc.currentMC.migrantFamily3_mc.onPress = function() {
		_root.talk_mc.talkScreen_mc._x = 500;
		_root.talk_mc.talkScreen_mc._y = 250;
		_root.talk_mc.talkScreen_mc.onEnterFrame = function() {
			checkClues(0);
			//trace("my clues are: "+clueValue);
			var myWords = "speak"+myLines;
			_root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[9].attributes[myWords];
			var myLength = allKnowing.allVoices[9].attributes.dialog;
			if (clueValue == true) {
				//trace("Truth!");
				if (myLines<myLength) {
					_root.talk_mc.talkScreen_mc.gotoAndStop(2);
				} else {
					_root.talk_mc.talkScreen_mc.gotoAndStop(1);
					delete this.onEnterFrame;
				}
				_root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
					soundFX("POP");
					myLines++;
					gameClue = true;
					pushClues();
				};
				_root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
					soundFX("POP");
					_root.talk_mc.talkScreen_mc._y = -1000;
					delete _root.talk_mc.talkScreen_mc.onEnterFrame;
					myLines = 1;
				};
				_root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
					soundFX("POP");
					_root.talk_mc.talkScreen_mc._y = -1000;
					myLines = 1;
				};
			} else if (clueValue == false) {
				//trace("BOGUS!");
				if (myLines<myLength) {
					_root.talk_mc.talkScreen_mc.gotoAndStop(2);
				} else {
					_root.talk_mc.talkScreen_mc.gotoAndStop(1);
					_root.talk_mc.talkScreen_mc.item_txt.text = "I'm sorry... better luck next time";
					delete this.onEnterFrame;
				}
				_root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
					soundFX("POP");
					_root.talk_mc.talkScreen_mc.gotoAndStop(1);
					//delete _root.talk_mc.talkScreen_mc.onEnterFrame;
					myLines = 2;
				};
```

```
                _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                    soundFX(“POP”);
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                    myLines = 1;
                };
                _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                    soundFX(“POP”);
                    gameClue = false;
                    pushClues();
                    _root.talk_mc.talkScreen_mc._y = -1000;
                    myLines = 1;
                };
            }
        };
    };
}
// Migrant Tent 4
if (myPerson == allKnowing.allVoices[10].attributes.item) {
    _root.movieClipLoader_mc.currentMC.migrantFamily4_mc.onRollOver = function() {
        this.onEnterFrame = function() {
            _root.speak_mc.speakMC._x = _xmouse
            _root.speak_mc.speakMC._y = _ymouse-10
        };
    };
    _root.movieClipLoader_mc.currentMC.migrantFamily4_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.speak_mc.speakMC._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.migrantFamily4_mc.onPress = function() {
        _root.talk_mc.talkScreen_mc._x = 500;
        _root.talk_mc.talkScreen_mc._y = 250;
        _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
            var myWords = “speak”+myLines;
            _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[10].attributes[myWords];
            var myLength = allKnowing.allVoices[10].attributes.dialog;
            if (myLines<myLength) {
                _root.talk_mc.talkScreen_mc.gotoAndStop(2);
            } else {
                _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                delete this.onEnterFrame;
            }
            _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                soundFX(“POP”);
                myLines++;
            };
            _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                soundFX(“POP”);
                _root.talk_mc.talkScreen_mc._y = -1000;
                delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                myLines = 1;
            };
            _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                soundFX(“POP”);
                _root.talk_mc.talkScreen_mc._y = -1000;
                myLines = 1;
            };
        };
    };
}
// Orange Grove
```

```
if (myPerson == allKnowing.allVoices[11].attributes.item) {
    _root.movieClipLoader_mc.currentMC.orangePerson_mc.onRollOver = function() {
        this.onEnterFrame = function() {
            _root.speak_mc.speakMC._x = _xmouse
            _root.speak_mc.speakMC._y = _ymouse-10
        };
    };
    _root.movieClipLoader_mc.currentMC.orangePerson_mc.onRollOut = function() {
        delete this.onEnterFrame;
        _root.speak_mc.speakMC._y = -1000;
    };
    _root.movieClipLoader_mc.currentMC.orangePerson_mc.onPress = function() {
        _root.talk_mc.talkScreen_mc._x = 500;
        _root.talk_mc.talkScreen_mc._y = 250;
        _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
            //trace(“my clues are: “+clueValue);
            var myWords = “speak”+myLines;
            _root.talk_mc.talkScreen_mc.item_txt.text = allKnowing.allVoices[11].attributes[myWords];
            var myLength = allKnowing.allVoices[11].attributes.dialog;
            if (myLines == 1) {
                checkClues(2);
                if (clueValue == true) {
                    //trace(“Truth!”);
                    if (myLines<myLength) {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                    } else {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        delete this.onEnterFrame;
                    }
                    _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                        soundFX(“POP”);
                        myLines++;
                    };
                    _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 1;
                    };
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        myLines = 1;
                    };
                } else if (clueValue == false) {
                    //trace(“BOGUS!”);
                    if (myLines<myLength) {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                    } else {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        _root.talk_mc.talkScreen_mc.item_txt.text = “I’m sorry... better luck next time”;
                        delete this.onEnterFrame;
                    }
                    _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        //delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 4;
                    };
                    _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        //delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 4;
                    };
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        myLines = 1;
                    };
                }
            }
            if (myLines == 2) {
                checkClues(4);
                if (clueValue == true) {
                    //trace(“Truth!”);
                    if (myLines<myLength) {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                    } else {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        delete this.onEnterFrame;
                    }
                    _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                        soundFX(“POP”);
                        myLines++;
                    };
                    _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 1;
                    };
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        myLines = 1;
                    };
                } else if (clueValue == false) {
                    //trace(“BOGUS!”);
                    if (myLines<myLength) {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                    } else {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        _root.talk_mc.talkScreen_mc.item_txt.text = “I’m sorry... better luck next time”;
                        delete this.onEnterFrame;
                    }
                    _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        //delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 4;
                    };
                    _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        //delete _root.talk_mc.talkScreen_mc.onEnterFrame;
                        myLines = 4;
                    };
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        myLines = 1;
```

```
                        };
                    }
                }
                if (myLines == 3) {
                    if (myLines<myLength) {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(2);
                    } else {
                        _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                        delete this.onEnterFrame;
                    }
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        myLines = 1;
                        finishGame(“end2MC”, allKnowing.allEnding[2].attributes.item);
                    };
                }
                if (myLines == 4) {
                    _root.talk_mc.talkScreen_mc.gotoAndStop(1);
                    _root.talk_mc.talkScreen_mc.item_txt.text = “I’m sorry... better luck next time”;
                    delete this.onEnterFrame;
                    _root.talk_mc.talkScreen_mc.close_mc.onRelease = function() {
                        soundFX(“POP”);
                        _root.talk_mc.talkScreen_mc._y = -1000;
                        myLines = 1;
                        finishGame(“end1MC”, allKnowing.allEnding[1].attributes.item);
                    };
                }
            };
        };
    }
};
```

Actionscript Code: Include “populateTown.as”

```
function populateTown(store1, store2, store3, store4, myTown, myClue) {
    _root.movieClipLoader_mc.currentMC.attachMovie(store1, “store1_mc”, 3);
    _root.movieClipLoader_mc.currentMC.store1_mc._x = 100;
    _root.movieClipLoader_mc.currentMC.store1_mc._y = 125;
    _root.movieClipLoader_mc.currentMC.store1_mc._xscale = 50;
    _root.movieClipLoader_mc.currentMC.store1_mc._yscale = 50;
    _root.movieClipLoader_mc.currentMC.attachMovie(store2, “store2_mc”, 2);
    _root.movieClipLoader_mc.currentMC.store2_mc._x = -75;
    _root.movieClipLoader_mc.currentMC.store2_mc._y = 75;
    _root.movieClipLoader_mc.currentMC.store2_mc._xscale = 37.5;
    _root.movieClipLoader_mc.currentMC.store2_mc._yscale = 37.5;
    _root.movieClipLoader_mc.currentMC.attachMovie(store3, “store3_mc”, 1);
    _root.movieClipLoader_mc.currentMC.store3_mc._x = -200;
    _root.movieClipLoader_mc.currentMC.store3_mc._y = 35;
    _root.movieClipLoader_mc.currentMC.store3_mc._xscale = 25;
    _root.movieClipLoader_mc.currentMC.store3_mc._yscale = 25;
    _root.movieClipLoader_mc.currentMC.attachMovie(store4, “store4_mc”, 4);
    _root.movieClipLoader_mc.currentMC.store4_mc._x = -280;
    _root.movieClipLoader_mc.currentMC.store4_mc._y = 100;
    _root.movieClipLoader_mc.currentMC.attachMovie(“rearModelT”, “modelT_mc”, 9);
    _root.movieClipLoader_mc.currentMC.modelT_mc._x = 300;
    _root.movieClipLoader_mc.currentMC.modelT_mc._y = 175;
    _root.movieClipLoader_mc.currentMC.attachMovie(“townSign_mc”, “townSign_mc”, 10);
    _root.movieClipLoader_mc.currentMC.townSign_mc._x = -170;
    _root.movieClipLoader_mc.currentMC.townSign_mc._y = 170;
    _root.movieClipLoader_mc.currentMC.townSign_mc.townSign_txt.text = myTown;
    //
    trace(myClue);
    //
    //
    _root.movieClipLoader_mc.currentMC.modelT_mc.onRelease = function() {
        soundFX (“BUZZER”)
        _root.talk_mc.talkScreen_mc.onEnterFrame = function() {
            _root.talk_mc.talkScreen_mc._x = (Stage.width/2)+50;
            _root.talk_mc.talkScreen_mc._y = (Stage.height/2)+100;
            _root.talk_mc.talkScreen_mc.gotoAndStop(2);
            _root.talk_mc.talkScreen_mc.item_txt.text = “Are you sure you want to leave?”;
            _root.talk_mc.talkScreen_mc.yes_mc.onRelease = function() {
                frameCount++;
                soundFX (“GO”)
                _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
                _root.talk_mc.talkScreen_mc._y = -1000;
                delete _root.talk_mc.talkScreen_mc.onEnterFrame;
            };
            _root.talk_mc.talkScreen_mc.no_mc.onRelease = function() {
                soundFX (“POP”)
                _root.talk_mc.talkScreen_mc._y = -1000;
                delete _root.talk_mc.talkScreen_mc.onEnterFrame;
            };
        };
    };
    _root.movieClipLoader_mc.currentMC.store1_mc.onRelease = function() {
        soundFX (“BUZZER”)
        myStore = store1+”_mc”;
        unPopulateTown(myStore, getClue, store1);
    };
    _root.movieClipLoader_mc.currentMC.store2_mc.onRelease = function() {
        soundFX (“BUZZER”)
        myStore = store2+”_mc”;
        unPopulateTown(myStore, getClue, store2);
    };
    _root.movieClipLoader_mc.currentMC.store3_mc.onRelease = function() {
        soundFX (“BUZZER”)
        myStore = store3+”_mc”;
        unPopulateTown(myStore, getClue, store3);
    };
    //
    _root.movieClipLoader_mc.currentMC.store4_mc.onRelease = function() {
        soundFX (“BUZZER”)
        _root.movieClipLoader_mc.currentMC.attachMovie(“wantGas_mc”, “wantGas_mc”, 100);
        _root.movieClipLoader_mc.currentMC.wantGas_mc._x = -100;
        _root.movieClipLoader_mc.currentMC.wantGas_mc._y = -100;
        _root.movieClipLoader_mc.currentMC.wantGas_mc.want_txt.text = “want gas?”;
        _root.movieClipLoader_mc.currentMC.wantGas_mc.gasPrice_txt.text = “$0.”+gasPrice();
        _root.movieClipLoader_mc.currentMC.wantGas_mc.yes_mc.onPress = function() {
            addGas();
            soundFX (“POP”)
            _root.movieClipLoader_mc.currentMC.wantGas_mc.unloadMovie();
        };
        _root.movieClipLoader_mc.currentMC.wantGas_mc.no_mc.onPress = function() {
            _root.movieClipLoader_mc.currentMC.wantGas_mc.unloadMovie();
            soundFX (“POP”)
        };
    };
}
unPopulateTown = function (myStore, getClue, whatStore) {
    _root.talk_mc.talkScreen_mc._y = -1000;
    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
    //
    alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
    _root.movieClipLoader_mc.currentMC.store1_mc.unloadMovie();
    _root.movieClipLoader_mc.currentMC.store2_mc.unloadMovie();
    _root.movieClipLoader_mc.currentMC.store3_mc.unloadMovie();
    _root.movieClipLoader_mc.currentMC.store4_mc.unloadMovie();
    _root.movieClipLoader_mc.attachMovie(myStore, “currentMC”, 1);
    _root.movieClipLoader_mc.currentMC.onEnterFrame = function() {
        getFood();
        getSupplies();
        peopleSpeak(whatStore);
        delete this.onEnterFrame;
    };
    _root.movieClipLoader_mc.currentMC.attachMovie(“exit_mc”, “exit_mc”, 2);
    _root.movieClipLoader_mc.currentMC.exit_mc._x = 240;
    _root.movieClipLoader_mc.currentMC.exit_mc._y = 200;
    _root.movieClipLoader_mc.currentMC.exit_mc.onPress = function() {
        _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        _root.talk_mc.talkScreen_mc._y = -1000;
        myLines = 1;
        soundFX (“EXIT”)
        attachClues(“unload”);
    };
    //trace(“Clue Value :”+getClue);
    /*if (getClue == true) {
    _root.movieClipLoader_mc.currentMC.attachMovie(“wantGas_mc”, “wantClue_mc”, 100);
    _root.movieClipLoader_mc.currentMC.wantClue_mc._x = -200;
    _root.movieClipLoader_mc.currentMC.wantClue_mc._y = -200;
    _root.movieClipLoader_mc.currentMC.wantClue_mc.want_txt.text = “want clue?”;
    _root.movieClipLoader_mc.currentMC.wantClue_mc.yes_mc.onPress = function() {
    gameClue = true;
    _root.movieClipLoader_mc.currentMC.wantClue_mc.unloadMovie();
    };
    _root.movieClipLoader_mc.currentMC.wantClue_mc.no_mc.onPress = function() {
    gameClue = false;
    _root.movieClipLoader_mc.currentMC.wantClue_mc.unloadMovie();
    };
    }*/
};
```

Actionscript Code: Include “random.as”

```
function myRandom(minVal, maxVal) {
    return minVal+Math.floor(Math.random()*(maxVal+1-minVal));
}
```

Actionscript Code: Include “testAmounts.as”

```
checkCash = function (itemCost, action) {
    if (action == “minus”) {
        if (totalCash<itemCost) {
            _root.movieClipLoader_mc.currentMC.cash_txt.text = “Broke!”;
            _root.movieClipLoader_mc.currentMC.cash_txt.textColor = “0X660000”;
        } else if (totalCash>itemCost) {
            totalCash = totalCash-itemCost;
            return totalCash;
        }
    }
    if (action == “add”) {
        totalCash = totalCash+itemCost;
        return totalCash;
    }
};
checkWeight = function (action) {
    if (totalWeight>=100) {
        if (puttputt == true) {
            attachMessage(“weight”, “Your car is overweight! Please try to limit the amount to under 100 pounds”);
            puttputt = false;
            overweight = true;
            weightCounter();
        } else {
            puttputt = true;
            overweight = false;
        }
    }
};
checkFood = function () {
    goodFoodObjects = new Array();
    extraFoodObjects = new Array();
    noFoodObjects = new Array();
    for (var j = 2; j<=(allKnowing.allPrices.length)-1; j++) {
        var myFood = allKnowing.allPrices[j].attributes.item;
        var foodCount = “item”+j;
        foodCount = foodSupplies[myFood];
        if (foodCount>0) {
            if (myFood == “flour” || myFood == “potato” || myFood == “vegetable” || myFood == “pork”) {
                goodFoodObjects.push(myFood);
                happiness++
            } else {
                extraFoodObjects.push(myFood);
                happiness=happiness+2;
            }
        }
        if (foodCount<=0) {
            if (myFood == “flour” || myFood == “potato” || myFood == “vegetable” || myFood == “pork”) {
                happiness--;
                noFoodObjects.push(myFood);
            } else {
                null;
            }
        }
    }
    minusFood();
};
minusFood = function () {
    for (var g = 0; g<=goodFoodObjects.length-1; g++) {
        var goodFood = goodFoodObjects[g];
        var goodItems = 1/goodFoodObjects.length;
        var myMembers = gamePlayerArray.length;
        var totalConsume = goodItems*myMembers;
        if (foodSupplies[goodFood]>=totalConsume) {
            foodSupplies[goodFood] = foodSupplies[goodFood]-totalConsume;
            addFood(goodFood, “minus”, totalConsume);
        } else {
            addFood(goodFood, “delete”, foodSupplies[goodFood]);
            noFoodObjects.push(goodFood);
            happiness--;
        }
    }
    for (var e = 0; e<=extraFoodObjects.length-1; e++) {
        var extraFood = extraFoodObjects[e];
        var extraItems = .2/extraFoodObjects.length;
        var myMembers = gamePlayerArray.length;
        var totalConsume = extraItems*myMembers;
        if (foodSupplies[extraFood]>=totalConsume) {
            foodSupplies[extraFood] = foodSupplies[extraFood]-totalConsume;
            addFood(extraFood, “minus”, totalConsume);
        } else {
            addFood(extraFood, “delete”, foodSupplies[extraFood]);
        }
    }
    if (noFoodObjects.length>=1) {
        puttputt = false;
        noFood = true;
        attachMessage(“food”, “You’re out of food! You need “+noFoodObjects+”. Please get to the nearest town quickly!”);
    } else {
        puttputt = true;
        noFood = false;
    }
};
```

Actionscript Code: Include “populateTents.as”

```
function populateTents(tent1, tent2, tent3, tent4, myClue) {
    _root.talk_mc.talkScreen_mc._y = -1000;
    _root.movieClipLoader_mc.currentMC.attachMovie(“tent1”, “tent1”, 1);
    _root.movieClipLoader_mc.currentMC.tent1._x = -125;
    _root.movieClipLoader_mc.currentMC.tent1._y = 150;
    _root.movieClipLoader_mc.currentMC.attachMovie(“tent2”, “tent2”, 2);
    _root.movieClipLoader_mc.currentMC.tent2._x = 200;
    _root.movieClipLoader_mc.currentMC.tent2._y = 150;
    _root.movieClipLoader_mc.currentMC.attachMovie(“tent3”, “tent3”, 3);
    _root.movieClipLoader_mc.currentMC.tent3._x = 500;
    _root.movieClipLoader_mc.currentMC.tent3._y = 150;
    _root.movieClipLoader_mc.currentMC.attachMovie(“tent4”, “tent4”, 4);
    _root.movieClipLoader_mc.currentMC.tent4._x = 800;
    _root.movieClipLoader_mc.currentMC.tent4._y = 150;
    //
    trace(myClue);
    _root.movieClipLoader_mc.currentMC.tent1.onRelease = function() {
        if (tent1 == myClue) {
            var getClue = true;
        } else {
            var getClue = false;
        }
        mytent = tent1+”_mc”;
        soundFX (“BUZZER”)
        unPopulateTown(mytent, getClue, tent1);
    };
    _root.movieClipLoader_mc.currentMC.tent2.onRelease = function() {
        if (tent2 == myClue) {
            var getClue = true;
        } else {
            var getClue = false;
        }
        mytent = tent2+”_mc”;
        soundFX (“BUZZER”)
        unPopulateTown(mytent, getClue, tent2);
    };
    _root.movieClipLoader_mc.currentMC.tent3.onRelease = function() {
        if (tent3 == myClue) {
            var getClue = true;
        } else {
            var getClue = false;
        }
        mytent = tent3+"_mc";
        soundFX ("BUZZER")
        unPopulateTown(mytent, getClue, tent3);
    };
    _root.movieClipLoader_mc.currentMC.tent4.onRelease = function() {
        if (tent4 == myClue) {
            var getClue = true;
        } else {
            var getClue = false;
        }
        mytent = tent4+"_mc";
        soundFX ("BUZZER")
        unPopulateTown(mytent, getClue, tent4);
    };
}
unPopulateTents = function (mytent, getClue, whatTent) {
    _root.talk_mc.talkScreen_mc._y = -1000;
    delete _root.talk_mc.talkScreen_mc.onEnterFrame;
    //
    peopleSpeak(whatTent);
    alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
    _root.movieClipLoader_mc.currentMC.tent1_mc.unloadMovie();
    _root.movieClipLoader_mc.currentMC.tent2_mc.unloadMovie();
    _root.movieClipLoader_mc.currentMC.tent3_mc.unloadMovie();
    _root.movieClipLoader_mc.currentMC.tent4_mc.unloadMovie();
    _root.movieClipLoader_mc.attachMovie(mytent, "currentMC", 1);
    _root.movieClipLoader_mc.currentMC.attachMovie("exit_mc", "exit_mc", 2);
    _root.movieClipLoader_mc.currentMC.exit_mc._x = 240;
    _root.movieClipLoader_mc.currentMC.exit_mc._y = 240;
    _root.movieClipLoader_mc.currentMC.exit_mc.onPress = function() {
        _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        _root.talk_mc.talkScreen_mc._y = -1000;
        myLines = 1;
        soundFX ("EXIT")
        //attachClues("unload");
    };
};
moveAmount = 10;
minLocX = Stage.width-1600;
function moveScreen() {
    _root.movieClipLoader_mc.currentMC.onEnterFrame = function() {
        checkTheMouse();
        trace("hello : "+_xmouse);
    };
}
checkTheMouse = function () {
    if (_xmouse<50) {
        moveImageRight();
    } else if (_xmouse>750) {
        moveImageLeft();
    } else {
        dontMove();
    }
};
moveImageRight = function () {
    Mouse.hide();
    leftArrow.leftArrowMC._x = _xmouse;
```

```
    leftArrow.leftArrowMC._y = _ymouse;
    _root.movieClipLoader_mc.currentMC._x = _root.movieClipLoader_mc.currentMC._x+moveAmount;
    if (_root.movieClipLoader_mc.currentMC._x>0) {
        _root.movieClipLoader_mc.currentMC._x = 0;
    }
};
moveImageLeft = function () {
    Mouse.hide();
    rightArrow.rightArrowMC._x = _xmouse;
    rightArrow.rightArrowMC._y = _ymouse;
    _root.movieClipLoader_mc.currentMC._x = _root.movieClipLoader_mc.currentMC._x-moveAmount;
    if (_root.movieClipLoader_mc.currentMC._x<minLocX) {
        _root.movieClipLoader_mc.currentMC._x = minLocX;
    }
};
dontMove = function () {
    Mouse.show();
    leftArrow.leftArrowMC._x = 0;
    leftArrow.leftArrowMC._y = -400;
    rightArrow.rightArrowMC._x = 0;
    rightArrow.rightArrowMC._y = -400;
};
```

Actionscript Code: Include “checkAmounts.as”

```
checkCash = function (itemCost, action) {
    if (action == “minus”) {
        if (totalCash<itemCost) {
            _root.movieClipLoader_mc.currentMC.cash_txt.text = “Broke!”;
            _root.movieClipLoader_mc.currentMC.cash_txt.textColor = “0X660000”;
        } else if (totalCash>itemCost) {
            totalCash = int(totalCash)-int(itemCost);
            return totalCash;
        }
    }
    if (action == “add”) {
        totalCash = int(totalCash)+int(itemCost);
        return totalCash;
    }
};
checkWeight = function (action) {
    if (totalWeight>=100) {
        if (puttputt == true) {
            attachMessage(“weight”, “Your car is overweight! Please try to limit the amount to under 100 pounds”);
            puttputt = false;
            overweight = true;
            weightCounter();
        } else {
            puttputt = true;
            overweight = false;
        }
    }
};
checkFood = function () {
    goodFoodObjects = new Array();
    extraFoodObjects = new Array();
    noFoodObjects = new Array();
    for (var j = 2; j<=(allKnowing.allPrices.length)-1; j++) {
        var myFood = allKnowing.allPrices[j].attributes.item;
        var foodCount = “item”+j;
        foodCount = foodSupplies[myFood];
        if (foodCount>0) {
            if (myFood == “flour” || myFood == “potato” || myFood == “vegetable” || myFood == “pork”) {
                goodFoodObjects.push(myFood);
                happiness++
            } else {
                extraFoodObjects.push(myFood);
                happiness=happiness+2;
            }
        }
        if (foodCount<=0) {
            if (myFood == “flour” || myFood == “potato” || myFood == “vegetable” || myFood == “pork”) {
                happiness--;
                noFoodObjects.push(myFood);
            } else {
                null;
            }
        }
    }
    minusFood();
};
minusFood = function () {
    for (var g = 0; g<=goodFoodObjects.length-1; g++) {
        var goodFood = goodFoodObjects[g];
        var goodItems = 1/goodFoodObjects.length;
        var myMembers = gamePlayerArray.length;
        var totalConsume = goodItems*myMembers;
        if (foodSupplies[goodFood]>=totalConsume) {
            foodSupplies[goodFood] = foodSupplies[goodFood]-totalConsume;
            addFood(goodFood, “minus”, totalConsume);
        } else {
            addFood(goodFood, “delete”, foodSupplies[goodFood]);
            noFoodObjects.push(goodFood);
            happiness--;
        }
    }
    for (var e = 0; e<=extraFoodObjects.length-1; e++) {
        var extraFood = extraFoodObjects[e];
        var extraItems = .2/extraFoodObjects.length;
        var myMembers = gamePlayerArray.length;
        var totalConsume = extraItems*myMembers;
        if (foodSupplies[extraFood]>=totalConsume) {
            foodSupplies[extraFood] = foodSupplies[extraFood]-totalConsume;
            addFood(extraFood, “minus”, totalConsume);
        } else {
            addFood(extraFood, “delete”, foodSupplies[extraFood]);
        }
    }
    if (noFoodObjects.length>=1) {
        puttputt = false;
        noFood = true;
        attachMessage(“food”, “You’re out of food! You need “+noFoodObjects+”. Please get to the nearest town quickly!”);
    } else {
        puttputt = true;
        noFood = false;
    }
};
```

Actionscript Code: Include “checkClues.as”

```
pushClues = function () {
    gameCluesArray.push(gameClue);
    trace(“my clues are: “+gameCluesArray);
};
checkClues = function () {
    trace(“my clues are: “+gameCluesArray);
};
```

Actionscript Code: Include “checkTotalTrip.as”

```
checkTotalMovement = function () {
    _root.movieClipLoader_mc.currentMC.onEnterFrame = function() {
        if (puttputt == true) {
            addMiles();
            moveCar(puttputt);
            penaltyCounter()
        }
        if (puttputt == false) {
            moveCar(puttputt);
        }

        if (noFood == true){
            deathCounter(noFood)
        }
        if (mileCounter == 80 | mileCounter == 510 | mileCounter == 775 | mileCounter == 1050 | mileCounter == 1220 ) {
            readMarker(mileCounter);
        }
        if (mileCounter == 40 | mileCounter == 120 | mileCounter == 200 | mileCounter == 250 | mileCounter == 360 | mileCounter == 400 |
        mileCounter == 480 | mileCounter == 540 | mileCounter == 650 | mileCounter == 720 | mileCounter == 800 | mileCounter == 850 |
        mileCounter == 960 | mileCounter == 1020 | mileCounter == 1080 | mileCounter == 1120 | mileCounter == 1200 | mileCounter ==
        1250) {
            getGas();
        }
        if (mileCounter == newDay) {
            calculateSupplies();
        }
        if (mileCounter == allKnowing.allCities[1].attributes.miles | mileCounter == allKnowing.allCities[2].attributes.miles | mileCounter ==
        allKnowing.allCities[3].attributes.miles | mileCounter == allKnowing.allCities[4].attributes.miles | mileCounter == allKnowing.allCities[5].
        attributes.miles | mileCounter == allKnowing.allCities[6].attributes.miles) {
            frameCount++;
            loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
        }
    };
};
```

Actionscript Code: Include “moveCar.as”

```
moveCar = function () {
    if (puttputt == true) {
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc.onEnterFrame = function() {
            //trace(_root.movieClipLoader_mc.currentMC._x);
            this._x += 20;
            if (this._x>=(this._width/2)) {
                this._x = 0;
            }
        };
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc.onEnterFrame = function() {
            //trace(this._x);
            this._x += 10;
            if (this._x>=(this._width/2)-10) {
                this._x = 0;
            }
        };
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc.onEnterFrame = function() {
            //trace(this._x);
            this._x += 5;
            if (this._x>=(this._width/2)-5) {
                this._x = 0;
            }
        };
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.modelT_mc.onEnterFrame = function() {
            //trace(this._x);
            this.wheel1_mc._rotation -= 5;
            this.wheel2_mc._rotation -= 5;
        };
    } else if (puttputt == false) {
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc1_mc.onEnterFrame = null;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc2_mc.onEnterFrame = null;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.mc3_mc.onEnterFrame = null;
        _root.movieClipLoader_mc.currentMC.tripViewer_mc.modelT_mc.onEnterFrame = null;
    }
};
```

Actionscript Code: Include “endGame.as”

```
finishGame = function (myEnd, endNum) {
    alphaFade(mx.transitions.easing.Strong.easeOut, 0, 100, 3, _root.movieClipLoader_mc);
    //_root.movieClipLoader_mc.currentMC.unloadMovie();
    _root.informationMC.interface_mc._visible = false;
    _root.movieClipLoader_mc.attachMovie(myEnd, “currentMC”, 1);
    //
    _root.movieClipLoader_mc.currentMC.onEnterFrame = function() {
        for (var endFam = 0; endFam<=_root.gameGenderArray.length; endFam++) {
            trace(endFam+” : “+gameGenderArray);
            FamMC = gameGenderArray[endFam];
            myFamMC = FamMC+”END”;
            newFamMC = “newFamMC”+endFam;
            myWorks = “work”+endFam;
            myTest = this.createEmptyMovieClip(myWorks, 1*endFam);
            gridClip = this[myWorks].attachMovie(myFamMC, “buttonClip”, 1+endFam);
            myTest._y = -(14*endFam)+15;
            trace(“Y Value : “+myTest._y);
            myTest._x = (22*endFam)-75;
        }
        delete this.onEnterFrame;
    };
    _root.movieClipLoader_mc.currentMC.attachMovie(“endTextMC”, “endText_mc”, 100);
    _root.movieClipLoader_mc.currentMC.endText_mc._x = -190;
    _root.movieClipLoader_mc.currentMC.endText_mc._y = 190;
    _root.movieClipLoader_mc.currentMC.endText_mc.endText_txt.text = endNum;
    //
    _root.movieClipLoader_mc.currentMC.game_btn.onPress = function() {
        frameCount++;
        _root.loadFrame(allKnowing.allMCs[frameCount].attributes.item, allKnowing.allMCs[frameCount].attributes.SWFloc);
    };
};
```

XML Code: “content.xml”

```
<setup>
    <movieClips>
        <clip item=”introduction1” SWFloc=”introduction_mc”/>
        <clip item=”welcome” SWFloc=”welcome_mc”/>
        <clip item=”familysetup” SWFloc=”familysetup_mc”/>
        <clip item=”supplies” SWFloc=”grocery_mc”/>
        <clip item=”materials” SWFloc=”fivedime_mc”/>
        <clip item=”oklahomacity” SWFloc=”oklahomacity_mc”/>
        <clip item=”route66_1” SWFloc=”route66_1_mc”/>
        <clip item=”Amarillo” SWFloc=”amarillo_mc”/>
        <clip item=”Amarillo” SWFloc=”amarilloTown_mc”/>
        <clip item=”route66_2” SWFloc=”route66_1_mc”/>
        <clip item=”Pie Town” SWFloc=”albuquerque_mc”/>
        <clip item=”Pie Town” SWFloc=”pieTownTown_mc”/>
        <clip item=”route66_3” SWFloc=”route66_2_mc”/>
        <clip item=”Flagstaff” SWFloc=”flagstaff_mc”/>
        <clip item=”Flagstaff” SWFloc=”flagstaffTown_mc”/>
        <clip item=”route66_4” SWFloc=”route66_3_mc”/>
        <clip item=”Needles” SWFloc=”needles_mc”/>
        <clip item=”Needles” SWFloc=”needlesTown_mc”/>
        <clip item=”route66_5” SWFloc=”route66_4_mc”/>
        <clip item=”San Bernardino” SWFloc=”sanbernardino_mc”/>
        <clip item=”San Bernardino” SWFloc=”sanbern_community_mc”/>
        <clip item=”route66_6” SWFloc=”route66_4_mc”/>
        <clip item=”migrant_encampment” SWFloc=”migrantCamp_mc”/>
        <clip item=”migrant_encampment” SWFloc=”tentcity_mc”/>
        <clip item=”orangegrove” SWFloc=”orangegrove_mc”/>
        <clip item=”scores” SWFloc=”ending_mc”/>
    </movieClips>
    <prices>
        <price item=”cash” price=”50” min=”0” max=”50” weight=””/>
        <price item=”gas” price=”15” min=”14” max=”17” weight=”150”/>
        <price item=”flour” price=”20” min=”18” max=”23” weight=”1”/>
        <price item=”potato” price=”25” min=”23” max=”28” weight=”1”/>
        <price item=”vegetable” price=”30” min=”28” max=”33” weight=”1”/>
        <price item=”pork” price=”40” min=”38” max=”43” weight=”1”/>
        <price item=”coffee” price=”35” min=”33” max=”38” weight=”1”/>
        <price item=”sugar” price=”45” min=”43” max=”48” weight=”1”/>
    </prices>
    <supplies>
        <supply item=”large pot” id=”largePot” minPrice=”310” maxPrice=”420” min=”100” max=”300” weight=”2”/>
        <supply item=”small pot” id=”smallPot” minPrice=”210” maxPrice=”330” min=”100” max=”200” weight=”1”/>
        <supply item=”large pan” id=”largePan” minPrice=”310” maxPrice=”440” min=”100” max=”300” weight=”2”/>
        <supply item=”small pan” id=”smallPan” minPrice=”210” maxPrice=”310” min=”100” max=”200” weight=”1”/>
        <supply item=”cleaning supplies” id=”soap” minPrice=”110” maxPrice=”150” min=”20” max=”90” weight=”1”/>
        <supply item=”family clothes” id=”clothes” minPrice=”210” maxPrice=”350” min=”100” max=”200” weight=”1”/>
        <supply item=”blankets and pillows” id=”blankets” minPrice=”220” maxPrice=”390” min=”100” max=”200” weight=”5”/>
        <supply item=”large water jug” id=”water” minPrice=”600” maxPrice=”800” min=”400” max=”600” weight=”80”/>
        <supply item=”small water jug” id=”water” minPrice=”500” maxPrice=”700” min=”300” max=”500” weight=”40”/>
    </supplies>
    <costs>
        <cost item=”flat tire” price=”2.50” penalty=”0”death =”0”/>
        <cost item=”flat tire” price=”2.50” penalty=”0”death =”0”/>
        <cost item=”flat tire” price=”2.50” penalty=”0”death =”0”/>
        <cost item=”flat tire” price=”2.50” penalty=”0”death =”0”/>
        <cost item=”flat tire” price=”2.50” penalty=”0”death =”0”/>
        <cost item=”overheated radiator” price=”5.00” penalty=”1” death =”0”/>
        <cost item=”overheated radiator” price=”5.00” penalty=”1” death =”0”/>
        <cost item=”blown head gasket” price=”10.00” penalty=”1” death =”0”/>
```

```
		<cost item=”got sick” price=”0”	penalty=”1”death =”0”/>
		<cost item=”got sick” price=”0”	penalty=”1”death =”0”/>
		<cost item=”got sick” price=”0”	penalty=”1”death =”0”/>
		<cost item=”got sick” price=”0”	penalty=”1”death =”0”/>
		<cost item=”died” price=”0” penalty=”1” death =”1”/>
	</costs>
	<cities>
		<city item=”Oklahoma City, Oklahoma” miles=”0” info=”Information on Oklahoma City” store1=”” store2=”” store3=”” store4=””/>
		<city item=”Amarillo” miles=”300” info=”Amarillo is one of the largest towns in the Texas Panhandle, one of the driest areas in the country. Amarillo was initially named Oneida, but would later change its name to Amarillo. This name was derived either from the yellow wild flowers that were abundant in the spring and summer seasons or after the yellow soil that lined the banks of the nearby lake and creek. The city is known as the ‘Yellow Rose of Texas.’ Amarillo is of note because it is the leading supplier of helium in the U.S. and the world.” store1=”grocery” store2=”sodafountain” store3=”fiveDime” store4=”gasStationStore” clue=”sodafountain” />
		<city item=”Pie Town” miles=”600” info=”Pie Town: America’s friendliest little town is located in west central New Mexico. Its name comes from Mr. Clyde Norman and his famous dried apple pie business that started in the 1920s. Word got around that the best pies were to be found at Pie Town and made this city a stop on many a migrant’s journey.” store1=”grocery” store2=”pieComm” store3=”fiveDime” store4=”gasStationStore” clue=”pieComm”/>
		<city item=”Flagstaff” miles=”900” info=”Flagstaff’s name comes from a Ponderosa pine tree that was made into a flagpole by a scouting party. This city’s first permanent settler arrived in 1876 and within ten years the population boomed to make Flagstaff the largest city between New Mexico and the west coast. It is home to the Lowell Observatory, where Pluto was discovered in 1930. Flagstaff’s relative closeness to the Grand Canyon has made this city a popular tourist destination.” store1=”grocery” store2=”fiveDime” store3=”soupkitchen” store4=”gasStationStore” clue=”soupkitchen”/>
		<city item=”Needles” miles=”1100” info=”Needles, the gateway to California, is located on the west bank of the Colorado River in Mojave Valley. It is named after the Needles, a group of pointed rocks on the eastern side of the river. Needles is known for extreme heat during the summer and will often set national daily temperature records. This city is a major railroad junction into California.” store1=”grocery” store2=”needlesComm” store3=”fiveDime” store4=”gasStationStore” clue=”needlesComm”/>
		<city item=”San Bernardino” miles=”1300” info=”San Bernardino is one of the oldest communities in the state of California. Founded in 1810, it was not largely populated until California achieved statehood in 1851. The first orange trees were brought to the city in 1857 and since that time, San Bernardino has been a major producer of oranges. Every April since 1911, the city hosts an Orange Festival. In 1948, the McDonald brothers opened the first McDonald’s Restaurant in San Bernardino.” store1=”sanBern” store2=”fiveDime” store3=”grocery” store4=”gasStationStore” clue=”sanBern”/>
		<city item=”Migrant Encampment” miles=”1325” info=”Information on Migrant Encampments” store1=”tent1” store2=”tent2”
		store3=”tent3” store4=”tent4” clue=”tent3”/>
	</cities>
	<markers>
		<marker item=”Route 66”	info=”Route 66 was the main highway connection to travel westward towards California for millions of people. Originating in Chicago, Illinois and terminating in Los Angeles, California, this route went through 8 states. The idea behind Route 66 was to connect rural communities to urban centers, thus allowing farmers to transport grains and produce to the market easier than by previously methods. A by-product of this route was its use as a migration route for impoverished farmers during the Dust Bowl Depression era of the late 1920-1930’s.”/>
		<marker item=”Civilian Conservation Corps” info=”The Civilian Conservation Corps (CCC) was a work relief program for young men established as part of President Franklin Delano Roosevelt’s New Deal programs. The CCC was one of the more popular New Deal programs as it employed young men to work on outdoor construction project like planting trees and building roads. Each CCC worker was paid $1 per day, which was a generous amount for that time. The main goal of this program was to get people back to work after the job losses due to the Great Depression.”/>
		<marker item=”The Works Public Administration”	info=”The Works Progress Administration (WPA) was one of the largest New Deal programs signed into law by President Franklin Delano Roosevelt. This program employed millions of people across the nation building many highways, dams and libraries. The WPA also employed many artists, musicians and actors to perform and create artistic pieces. Through this program, in addition to all the building, 225,000 concerts and 475,000 works of art were sponsored.”/>
		<marker item=”The NRA and AAA Acts” info=”Though strong in idea, the National Recovery Administration (NRA) and the Agricultural Adjustment Act (AAA) were two New Deal programs that failed miserably. The NRA aimed to promote fair competition and set competitive wages for workers. The AAA aimed to set fair prices on crops. The Supreme Court ruled that both of these programs were unconstitutional due to the price fixing that occurred which actually hurt competition.”/>
		<marker item=”Migrant Hardships” info=”The families making the journey out west often endured many trials as they tried to adjust to their new environments. They were singled out for ridicule, often being called ‘Okies’ as a reference of them being from Oklahoma. Due to the limited amount of work, migrant families often competed against each other for the low paying jobs that were available. Many workers were still unable to make ends meet. These families were often pushed into refugee camps where the stigma of being an ‘Okie” was made worse.”/>
	</markers>
	<voices>
		<voice item=”grocery” speak1=”Hello there. Please select an item of food to buy” dialog=”1”/>
		<voice item=”fiveDime” speak1=”Hello there. Please select an item to buy or sell” dialog=”1”/>
		<voice item=”pieComm” speak1=”Welcome to Pie Town. Would you like a map?” speak2=”here’s the map. Have a safe trip” dialog=”2”/>
		<voice item=”needlesComm” speak1=”Welcome to Needles. Would you like to register to work in California?” speak2=”Would you like the
```

```
paperwork you will need to register” speak3=”You can drop it off once you get to San Bernardino” dialog=”3”/>
        <voice item=”sodafountain” speak1=”Welcome to Amarillo. Are you going to California?” speak2=”Could you drop a letter off to my son in San Bernardino?” speak3=”You can’t miss him! He’s probably the only one with red hair!” dialog=”3”/>
        <voice item=”soupkitchen” speak1=”Welcome to Flagstaff. Would you like some food?” speak2=”I see you are in need of some help. Would you like some information on a job in California” speak3=”Once you get to the Orange Grove, say that Wilma sent you” dialog=”3”/>
        <voice item=”sanBern” speak1=”Welcome to San Bernardino. Do you have the paperwork to work in California” speak2=”Here’s your workers registration card. Go to the migrant camp outside of town to look for work” dialog=”2”/>
        <voice item=”tent1” speak1=”It’s tough to find a decent wage out here when the farmers only care for themselves!” dialog=”1”/>
        <voice item=”tent2” speak1=”Let me tell you, we can’t make enough to eat! “ dialog=”1”/>
        <voice item=”tent3” speak1=”You stopped in Amarillo? Did you get a letter from my dad at the soda fountain?” speak2=”Thanks! Hey, I think I might just know where that orange grove you’re looking for is...” dialog=”2”/>
        <voice item=”tent4” speak1=”5 cents a bushel for tomatoes? How do the farmers they expect us to make ends meet?” dialog=”1”/>
        <voice item=”orangeGrove” speak1=”I hope your journey went well. Do you have the handbill I sent out?” speak2=”Excellent! Do you have a workers registration card?” speak3=”Come on inside. Your family are hired!” dialog=”3”/>
    </voices>
    <ending>
        <end item=”I’m sorry, your family didn’t quite get there... better luck next time.”/>
        <end item=”I’m sorry, your family didn’t quite make it. During the time of the Great Depression, people could rely on the people they met and befriended. People facing similar hardships often foster a feeling of helping the greater good for all. You missed out on some valuable information along the way from people you could have met, and it has prevented you from landing the promise of a better life in California.”/>
        <end item=”Congratulations, your family made it! During the time of the Great Depression, people could rely on the people they met and befriended. People facing similar hardships often foster a feeling of helping the greater good for all. You managed to gather some valuable information along the way from people you have met, and this ultimately has afforded you the promise of a better life in California.”/>
    </ending>
</setup>
```

XML Code: “quiz.xml”

```
<interactionTwo>
    <content image=”fdrMC” question=”This man, the nations 32nd president, enacted programs like the New Deal to assist the country in relieving the economic hardships due to the Great Depression” answer1=”Frankiln Delano Roosevelt” answer2=”Winston Churchill” answer3=”Herber Hoover” answer4=”Woody Guthrie” respond=””/>
    <content image=”dustBowlMC” question=”Crop failures, drought and clouds of debris, gave the American Midwest during the 1920s and 1930s the nickname” answer1=”The Dust Bowl” answer2=”The Outback” answer3=”The Heartland” answer4=”Tornado Alley” respond=””/>
    <content image=”cccMC” question=”This program, put 2.5 million unmarried men to work maintaining and restoring forests, beaches, and parks. Workers earned only $1 a day but received free board and job training.” answer1=”Civilian Conservation Corps” answer2=”National Recovery Act” answer3=”Social Security Act” answer4=”Agricultural Adjustment Act” respond=””/>
    <content image=”hooverMC” question=”This dam, helped develop the American Southwest into a viable economy, providing jobs, and a stable source of electricity and water to the developing region” answer1=”Hoover Dam” answer2=”Grand Coulee Dam” answer3=”Bonneville Dam” answer4=”Chickamauga Dam” respond=””/>
    <content image=”migrantMC” question=”Farmers, left this region to head to the more productive agricultural areas of the western United States in hopes of better opportunities to support their families” answer1=”The Great Plains” answer2=”Rocky Mountains” answer3=”Southeast US” answer4=”Northeast US” respond=””/>
</interactionTwo>
```

XML Code: “soil.xml”

```
<interactionOne>
    <sectionOne>
        <newItem title=”Prairie Grasses” desc=”Before western farming practices took root in the midwest, a natural ecosystem of prairie grasses dominated the landscape.”/>
        <newItem title=”Prairie Grasses” desc=”Perennial grasses, through the cycle of growth and decay provide a give and take symbiosis with the soil.”/>
        <newItem title=”Prairie Grasses” desc=”Grasses and other plants of the prairie system dig deep and interconnected root systems, preventing moisture from escaping, and soil from eroding.”/>
    </sectionOne>
    <sectionTwo>
        <newItem title=”Row Crops” desc=”With the arrival of agriculture, the natural grasses were removed, and in their place farm crops rose up.”/>
        <newItem title=”Row Crops” desc=”Crops like corn, soybeans and cotton are not perennial. They require a large amount of nutrients from
        the soil, while not returning many in return”/>
        <newItem title=”Row Crops” desc=”Row crops also have weak root systems, allowing the soil to erode easily”/>
    </sectionTwo>
    <crops>
        <newItem title=”Corn” desc=”Corn, though a good crop, demands a large amount of water and nutrients to grow effectively.”/>
        <newItem title=”Cotton” desc=”Cotton is the most demanding of crops, requiring a large amount of nutrients, stripping an already parched soil.”/>
        <newItem title=”Soybeans” desc=”Soybeans are less demanding on the soil to grow, though the overall profit margin is low.”/>
    </crops>
</interactionOne>
```

XML Code: “hoover.xml”

```
<interactionThree>
    <items>
        <newItem title=”railroad” itemMC=”railroadMC” phrase=”Get supplies and workers to the job site before starting on construction”/>
        <newItem title=”tunnels” itemMC=”tunnelsMC” phrase=”Add tunnels to channel water before starting excavation on the canyon walls and floor”/>
        <newItem title=”coffer dam” itemMC=”cofferMC” phrase=”Add coffer dam to divert water to the tunnels so you can start excavation”/>
        <newItem title=”carve canyon” itemMC=”carveMC” phrase=”Carve and prepare canyon walls before the start of construcion on the Hoover Dam”/>
        <newItem title=”concrete” itemMC=”concreteMC” phrase=”Pour concrete for the Hoover Dam”/>
        <newItem title=”refrigerate” itemMC=”fridgeMC” phrase=”Refridgerate concrete to prevent the Hoover Dam from cracking”/>
        <newItem title=”turbines” itemMC=”powerMC” phrase=”Add electric turbines for power generation”/>
        <newItem title=”pump houses” itemMC=”excavateMC” phrase=”Pump water to turbines for power generation”/>
    </items>
    <descriptions>
        <desc item=”Hoover Dam” words=”The Hoover Dam forms Lake Meade, a large reservoir of water than can be drawn upon for hyrdo-electric power, irrigation and drinking water.”/>
        <desc item=”Refridgeration Plant” words=”The refridgeration plant serves to cool the concrete of the Hoover Dam. Without this plant, the dam would crack under heat stress.”/>
        <desc item=”Hydro-Electric Plant” words=”The hydro-electric plant generates power by passing running water past spinning turbines, thus generating power.”/>
        <desc item=”Pump Houses” words=”Pump houses work to force water down a network of pipes that spin the electric genereating turbines at the hydro-electric plant.”/>
        <desc item=”Railroad” words=”In order to build the Hoover Dam, a railroad had to be constructed to ship materials and a labor force to the work site.”/>
    </descriptions>
</interactionThree>
```

XML Code: “questions.xml”

```
<interactionTwo>
    <content question=”This man was the 32nd president. He enacted a series of legislation called The New Deal” answer1=”Frankiln Roosevelt” answer2=”Winston Churchill” answer3=”Huey Long” image1=”images/fdr.jpg” image2=”images/churchill.jpg” image3=”images/long.jpg”/>
    <content question=”This woman was the first woman to hold a presidential cabinet post, serving as Secretary of Labor “ answer1=”Frances Perkins” answer2=”Eleanor Roosevelt” answer3=”Margaret Sanger” image1=”images/perkins.jpg” image2=”images/eleanor.jpg” image3=”images/sanger.jpg”/>
    <content question=”This man was a famous musician at the time, composing songs about the Dust Bowl and Family hardships” answer1=”Woody Guthrie” answer2=”Glenn Miller” answer3=”Bing Crosby” image1=”images/guthrie.jpg” image2=”images/miller.jpg” image3=”images/crosby.jpg”/>
    <content question=”This man was the 31st president. His hands off approach during the Great Depression led to economic problems”
answer1=”Herbert Hoover” answer2=”Wendell Willkie” answer3=”Alfred Smith” image1=”images/hoover.jpg” image2=”images/willkie.jpg”
image3=”images/smith.jpg”/>
    <content question=”This woman was a famous aviatrix, being the first woman to fly across the Atlantic solo. “ answer1=”Amelia Earhart” answer2=”Georgia O’Keeffe” answer3=”Babe Zaharias” image1=”images/earhart.jpg” image2=”images/okeeffe.jpg” image3=”images/zaharias.jpg”/>
</interactionTwo>
```

Migration During the Dust Bowl

Bibliography:
Books

Cayton, Andrew R.L., Perry, E.I. & Winkler, A.M. America: Pathways to the Present
Upper Saddle River, N.J., Pearson Prentice Hall, 1998

Ganzel, Bill. Dust Bowl Descent
Lincoln, Nebraska. University of Nebraska Press, 1984

Heyman, T., Phillips, S.,& Szarkowski, J. Dorothea Lange: American Photographs
San Fransisco, California: Chronicle Books, 1994

Singleton, Jeff. Unemployment Relief and the Welfare State in the Great Depression
Westport, Connecticut: Greenwood Press, 2000

Stein, Walter J. California and the Dust Bowl Migration
Westport, Connecticut: Greenwood Press, 1973

Steinbeck, John. The Grapes of Wrath
New York: Knopf: Distributed by Random House, 1993

Bibliography:
Audio/Video

Lorentz, P. (Writer/Director). (1936). *The Plow That Broke the Plains* [Motion picture]. United States: Resettlement Administration.

Steinbeck, John., & Johnson, Nunnally. (Writers), & Ford, J. (Director). (1940). *The Grapes Of Wrath* [Motion picture].
United States: Twentieth Century-Fox Film Corporation

Migration During the Dust Bowl

Bibliography:
Websites

Best of History Web Sites, compilation of web site resources: (n.d).
Retrieved September 28, 2005
http://www.besthistorysites.net/USHistory_GreatDepression.shtml

California History Online, “The Great Depression: California in the Thirties(n.d.)”.
Retrieved December 12th, 2005
http://www.californiahistory.net/depres_frame_mainmenu.htm

Cunfer, J., Mr. (2004, August 19). The Dust Bowl. Retrieved January 15, 2006, from EH.net Encyclopedia Web site: http://www.eh.net/encyclopedia/article/Cunfer.DustBowl

EyewitnesstoHistory.com, Snapshots: America in the 1930’s (n.d.).
Retrieved December 12, 2005.
http://www.eyewitnesstohistory.com/snprelief.htm

Franklin Delano Roosevelt Presidential Library(n.d). Retrieved Sept. 30, 2005
http://www.fdrlibrary.marist.edu/

History, Arts and Libraries, Lesson Plans for the Great Depression(2001-2005).
Retrieved December 14, 2005
http://www.michigan.gov/hal/0,1607,7-160-17451_18670_18793-53530--,00.html

InflationData.com, Historical Consumer Price Index(2003-2004)
Retrieved, January 30, 2006
http://inflationdata.com/inflation/Consumer_Price_Index/HistoricalCPI.aspx?rsCPI_currentPage=5

Kingwood College Library, “American Cultural History, 1930-1939
(July 7th,2001)”. Retrieved December 12, 2005
http://kclibrary.nhmccd.edu/decade30.html

Library of Congress(n.d). Retrieved September 30, 2005
http://www.loc.gov/

Library of Congress, The American Folklife Center(n.d).
Retrieved September 30, 2005
http://www.loc.gov/folklife/

Library of Congress, American Life Histories, Manuscripts from 1936-40(n.d).
Retrieved September 30, 2005
http://memory.loc.gov/ammem/wpaintro/wpahome.html

Library of Congress, Voices From the Dust Bowl: The Charles L. Todd and
Robert Sonkin Migrant Worker Collection 1940-1940(January 8, 1998).
Retrieved December 6, 2005
http://memory.loc.gov/ammem/afctshtml/tshome.html

Bibliography:
Websites

Modern American Poetry, “The Great Depression (n.d.).”
Retrieved December 13th, 2005
http://www.english.uiuc.edu/maps/depression/depression.htm

The National Archive(n.d). Retrieved September 28, 2005
http://www.archives.gov/

The New Deal Network (2003). Retrieved December 13th, 2005
http://newdeal.feri.org/

New York Social Studies Resource Guide and Core Curriculum,
(March 10, 2006). Retrieved March 25th, 2006
http://www.emsc.nysed.gov/ciai/socst/ssrg.html

Public Broadcasting Service(n.d). Retrieved September 28, 2005
http://www.pbs.org/

PBS, The American Experience, “Surviving the Dust Bowl” (n.d).
Retrieved September 28, 2005
http://www.pbs.org/wgbh/amex/dustbowl/tguide/index.html

PBS, The American Experience, “The Presidents: FDR”(n.d).
Retrieved September 28, 2005
http://www.pbs.org/wgbh/amex/presidents/32_f_roosevelt/index.html

PBS, People’s Century, “Breadline” (n.d). Retrieved September 28, 2005
http://www.pbs.org/wgbh/peoplescentury/episodes/breadline/

Weedpatch Camp(n.d.) Retrieved December 26, 2005
http://www.weedpatchcamp.com/

Wessels Living History Farm, Farming in the 1930’s (n.d.)
Retrieved December 13, 2005
http://www.livinghistoryfarm.org/farminginthe30s/farminginthe1930s.html

Bibliography: Images

Chambers, G.R. (1909). ***San Bernardino, California, city and village*** [Image file]. Retrieved February 10, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/pan.6a17556

Delano, J. (1944). ***Amarillo, Texas, general view, Santa Fe R.R. trip*** [Image file]. Retrieved February 10, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/fsac.1a34724

Delano, J. (1944). ***Mojave Desert country, crossed by the Santa Fe R.R., Cadiz, Calif.*** [Image file]. Retrieved February 10, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/fsac.1a34762

U.S. Farm Administration. (1935). ***Prowers County, Colorado. Dust storm.,*** [Image file]. Retrieved February 10, 2006, from Library of Congress Web site: http://memory.loc.gov/pnp/fsa/8e03000/8e03000/8e03000u.tif

Lange, D. (1936). ***Destitute pea pickers in California. Mother of seven children. Age thirty-two. Nipomo, California*** [Image file]. Retrieved February 10, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/ppmsca.12883

Lange, Dorothea. ***Migrant Farmer***. Photograph. 1930s. Dorothea Lange: American Photographs. By Heyman, T., Phillips, S.,& Szarkowski, San Fransisco: Chronicle Books, 1994. 111.

Leo, R. (1939). ***Corn near Muskogee, Oklahoma*** [Image file]. Retrieved February 10, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/fsac.1a34087

Leo, R. (1940). ***Jim Norris and wife, homesteaders, Pie Town, New Mexico*** [Image file]. Retrieved April 6, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/fsac.1a34109

Leo, R. (1940). ***Main Street, Pie Town, New Mexico*** [Image file]. Retrieved April 6, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/fsac.1a34148

West Coast Art Co. (1909). PAN US GEOG - Arizona no. 58 (E size) [Data file]. Retrieved April 6, 2006, from Library of Congress Web site: http://hdl.loc.gov/loc.pnp/cph.3c22050