Final Project: Why did the chicken cross the road?  
(50 exam points)

The final project will consist of two parts.

(I will add hints as the semester progresses, so do check this copy frequently.)

Programming Part (30 points) Due Midnight April 29th

In this project, the chicken (Animals) has a real sweet-tooth and crosses the road (City/Road) to eat the gumdrops (Local Gallery/Kitchen/Food). Write an animation where the player guides the chicken across the road to get to the gumdrops. Cars should move in both directions as the chicken tries to cross to where the gumdrops are located. Use arrow keys to make the chicken jump left, right, forward, and back. Use the space bar to have the chicken peck at the gumdrop. When the chicken is close enough to the gumdrop and pecks, the gumdrop should disappear.

If the chicken gets hits by a vehicle, the game is over (squish!) The game continues as long as the chicken has not managed to peck all the gumdrops and has not been squished. If the player pecks all the gumdrops and crosses the road, the player wins the game. Signal the success by making the 3D text "You Win" appear.

Allow the user to play again by asking, "Play again?"

Assign a scoring system of your choice and keep track of the running score for all games. Display the score on the upper right hand corner.

Have at least three cars (of different colors) of the same user defined class. Have at least three different gumdrops of different colors.

The car's action (moveFlip) must be a car method. Since you want all cars to duplicate the same actions and keep track of when they need to flip, create a user defined class "highwayCar".
First get one class doing what you want. Then create other instances of that class. This is done as follows:

- Rename the class by right clicking on the class and giving it a new name (say, "highwayCar")
- Right click on the "highwayCar" and select "save Object". It will ask you where you want to store it. Pick a place you can remember.
- You can either copy an instance of your first car OR import it from the location you stored it. Importing is done via File/Import. The new instances of highwayCar will have the variables and methods of the first.

**Division of points:**
1. (2 points) Setup
2. (5 points) Chicken pecking/jumping methods controllable by user.
3. (3 points) Gumdrop disappearing when pecking is close enough
4. (3 points) Cars continuously moving along the road in both directions. Cars defined as a user define class.
5. (3 points) Cars squishing chicken if get too close. Create a squish method for the chicken which puts him into a "pose" in which his body parts are scattered. You must use "capture pose" for full credit.
6. (1 point) Keeping score over multiple games.
7. (1 point) Identifying when the chicken has won
8. (1 point) Asking user if he/she wants to repeat the game.
9. (3 points) Restarting the game.
10. (3 points) Doing something extra of your own choosing. Identify the extra component via comments.
11. (5 points) Overall design, use of methods, parameters, variable names, good logic, good use of comments.

**Hints:**
1. Make sure the chicken actions are "chicken" methods.
2. I had gumdrop variables which kept track of whether I had already eaten a gumdrop – so I knew when to stop.
3. I had a world variable "totalPoints"
4. Make sure your gumdrops are on the ground – otherwise the chicken doesn’t have a chance. In a 2D rendering, it is hard to tell whether things are on the ground or floating. I used the "pointOfView/position" property of the object to help in the positioning.
5. I wanted the camera to move with the chicken, but it was too distracting to move the camera every time the chicken moved. I used a "chicken" level variable "ctMoves" which I incremented every time the chicken moved. Every ten moves, I had the camera "move toward" the chicken. Feel free to develop other techniques for getting the camera to move closer to the action. You could even have the user control camera motion with an event, but
you will need to inform the user what he/she is supposed to do. This communication with the user can be done with a billboard that disappears after the game begins.

7. Because almost all of the action happens with button-click events, it is a bit tricky to have other types of control. In "my first method", I had a while loop that said something like:
   while there are still gumdrops
       move cars
       show score
   make the "you win" sign visible.

**Project Quiz (20 points)**
During our regular final examination period (11:30 a.m., Wednesday in Engr 103), you will have a 20 question multiple choice examination over algorithm design in Alice. You will be quizzed on basic terminology and algorithm design. This is required for all students.

**Comprehensive Final (90 points)**
During the last hour and a half of our regular final examination period (11:30 a.m., Wednesday in Engr 103), you will have a 90 point examination over the material of the first two exams. This is optional for all students. Unless you contact the instructor in writing on or before April 24th, there will NOT be an examination prepared for you and you will be unable to take it. The score on this exam will replace the lower of the two regular exams. Yes, you can do worse.