Assignment #6 – Snake

Date assigned: Tuesday, March 30, 2010
Date due (Part I): Tuesday, April 13, 2010
Points: 50
Date due (Part II): Tuesday, April 20, 2010
Points: 25
Group Presentation: 10

You are to implement a simple version of the game "Snake." Details and history of the game can be found at: **http://en.wikipedia.org/wiki/Snake_(video_game).**

I. Rules for the game (Part I Implementation):

1. A snake consisting of four parts is placed on the center of the screen.
2. The snake initially begins moving north when looking at the screen.
3. The snake moves at a constant speed.
4. The snake can only move north, south, east, and west using the arrow keys.
5. The snake moves on the screen by adding a head piece and subtracting a tail piece.
6. Along with the snake, an apple is to appear at a random location on the screen. There is to be exactly one visible apple on the screen at any point in time.
7. When the snake eats the apple, the snake gets one part larger.
8. The snake starts out at a slow pace and increases in pace for each five apples eaten.
9. Each time an apple is eaten, the score counter is incremented by 1; however, for each time the snake pace increases, the counter value is incremented by 1. So the first five apples count 1, the next five apples count 2, and so on.
10. The game continues until the snake dies. The snake dies by running into itself or by running into a wall.

II. Rules for the display (Part I Implementation):

1. The display needs to include:
    a. a simple border
    b. initially, the four part snake in the middle of the screen oriented north
    c. the word **Score:** at the bottom of the display
    d. the score total (number of apples eaten) to the right of Score:

III. High Scores (Part II Implementation):

1. You are to keep all scores of the game in a database. Each score value is to have an associated and date and high score.
2. A separate screen is to display the top scores from the highest score to the lowest score. Each line of score data is to show the date and score. You will get to this screen through a menu option.

IV. Game State (Part II Implementation)

1. A player playing a game of Snake is to be able to leave the game at any time and come back exactly where they left off.

V. Extra Credit (Part II Implementation)

1. Incorporate non-obnoxious sounds for each of the following:
    a. each time the snake moves
    b. when the snake eats an apple
    c. when the snake dies
2. As always, you can only receive the extra credit if your entire project is some kind of A (i.e. 90% or higher). Further, you are to decide whether you want 4 points of extra credit on the next exam, 6 points of extra credit on the next quiz, or 6 points of extra credit on this assignment.

Group Presentation on Thursday, April 1:

Each group is to get together and design the game as described above. Your design is to include:

1. Each class
    a. object hierarchy/implemented interfaces
    b. the class's public interface
    c. what class is responsible for what functionality
    d. what class is responsible for what data
2. Each thread
    a. what threads will be needed
    b. what will each thread do

Only one individual is to present for the entire group. You will most likely change the design as you begin to implement the project. Include in your program documentation the changes from your original design and why you made them. Giving a PowerPoint presentation is fine and preferable.

Notes:

1. There are to be enough specifics so the class knows pretty much how your game will be implemented and how your game will work.

2. Each group is to present their design on Thursday, April 1. The group order will be in the order listed at:
http://zeus.cs.pacificu.edu/ryand/cs360android/2010/Lectures/AndroidGroups.html

3. Everyone in the group will receive the same grade. You will receive a grade of 0 if you do not meet with your group OR you do not show up for class during the presentation. Answer any email from a group member IMMEDIATELY.

4. The minimum presentation time for each group is 7 minutes and the maximum time is 10 minutes. The presentation grade will be based on completeness, correctness, and clearly articulating your design to the class.

5. The design is a group design; however, the implementation is individual.

Goals for Assignment #5:

1. Work with a SurfaceView for doing simple 2D animation
2. Understand threads and how they work
3. Understand the details & complexities of saving a game state

Project Specifics

1. Save your Android project in a project folder called PUNetIDSnake. Then drop the entire folder into the CS360AndroidDrop folder by 1pm on the day in which the assignment is due.

2. You must use a SurfaceView for a draw surface and a draw thread that draws to the canvas.

3. Your program is to work in landscape mode. Further, you are to bring me a working version of your program on the Droid for both parts of this assignment.

4. Sprites are to be 20x20 in size.

5. The game must include some kind of visible border.

6. Your code is to be written using the development tools specified in the syllabus. Remember, I will be testing your app in a Windows emulator.

7. If you come to me with a question regarding your solution, I will have you load your project onto a machine in the CS lab. I will not look at your code on your computer or on paper as it just takes me too long to get at the problem. Further, I want you to bring in your textbook and lecture notes in case I want you to look up something. Remember, I'm not just a tell you the answer guy. Make sure you understand how to use the developer tools and that you can run your program on the emulator.

8. If you want help with a compiler error, you must be able to tell me exactly what statement you put in your code that caused the error and be able to isolate the error. If you have typed in a bunch of code and have not tested your code as you've gone along, I'm not going to help you sort out the mess. You've been warned!!

9. Print out all code generated for this assignment. Do not print out any XML code. Remember, your printed copy is also due by 1pm on the day in which the assignment is due. Print the main Activity code first.