Assignment 4: Super Mario Brothers

June 2017

1 General Instructions

1. **The assignment is to be done in groups of 2 people.** You may also choose to work solo.
2. We will be using GitHub for version control. Make sure to regularly commit your changes to your repository, rather than making a single commit at the end of the assignment. The commit history of the repository will be checked to verify the same.
3. The **deadline** for Assignment 2 is **11:59 p.m. on June 9th, 2017**. The latest commit in your GitHub repository, as of 11.59 p.m. on the 9th of June will be considered as your final submission.
4. Academic dishonesty is unacceptable and will not be tolerated in this course. To compare the code committed by the teams, an automatic system called MOSS will be used for determining the similarity of the programs.

2 Assignment Description

This assignment requires you to develop an Android game that mimics the classic game **Super Mario Brothers**. You can read more about the game at **`http://www.mariowiki.com/Super_Mario_Bros.`**.

We have made the following simplifying changes to the game:

1. There need not be any sounds.
2. You only need **3 levels**.
3. You can redesign the AI behaviors of the opponents as you wish.

Here are the basic points of the game (we leave lots of implementation choices to you):

1. The game is side-scrolling. This means that the characters onscreen are viewed from a side-view camera angle. Your game must make the onscreen characters move from the **right to left side**, like a mirror image of the original game.
2. The game must run in **Landscape mode**, unlike the Candy Crush and DigDug games which were Potrait oriented.
3. The original game has several items. But, you need to implement only the following:
   - Coins: They are worth 200 points and are very common.
   - Super Mushroom: They are worth 1000 points, and are less common. It gives Mario the ”Super” form.
   - Fire Flower: They are also worth 1000 points and they give Mario the ability to shoot fireballs, by transforming him into ”Fire Mario” form.
4. Mario has different transformations. Mario transforms between those versions by touching an enemy or obstacle, or by acquiring a particular type of item. You need to implement the ”Super” and ”Fire Mario” versions, apart from the version of Mario at the beginning of the game. The different transformations of Mario and the details of when they occur is listed **here**.

5. Among the many enemies that Mario has, you have to implement at least 3 types. The possible enemies are:

   - Buzzy Beetle : It is small turtle that hides in its shell when jumped on, and it is immune to fireballs.
   - Bloober: It is a squid-like sentry that persistently pursues the player.
   - Koopa Paratroopa: They are small turtles with wings. Hostile green ones jump towards the player, and breezy red ones fly back and forth, or up and down.
   - Piranha Plant: It is a carnivorous plant that lives in pipes. It rises up trying to hit Mario and retreats. If Mario is near, it won't rise up.

   Descriptions of the enemies are available in more detail in **this link**.

6. Mario should be able to jump. He should also be able to break certain types of blocks.

7. Mario should have 3 lives. Keep a running score and also keep track of the number of lives.

8. Artwork does not matter in this class. But, Mario should be animated in some fashion.

9. The different levels should be controlled by a data structure. The placement of items should be controlled by the data structure. There should be no hard coded checks in the game engine about what level you are in (other than the code to load the correct data structure for the level).

10. You will have to develop touch controls.

3 Hints

The Android development kit documentation is your friend. Reference it to figure out how the entire thing works.
The URL is `http://developer.android.com/reference/packages.html`. A copy also gets installed in your Android installation.

3.1 How do I change the orientation of the screen?

1. The method `void setRequestedOrientation (int requestedOrientation)` can be used to set the orientation of the activity.

2. The value of requestedOrientation can be set to `SCREEN_ORIENTATION_LANDSCAPE`, to render the game in landscape fashion.

4 No-Copy Policy

The assignment must contain the following statement in the file **"statement.txt"**:

We, your name and your partner, hereby certify that the files we submitted represent our own work, that we did not copy any code from any other person or source, and that we did not share our code with any other students.

NOTE: It is acceptable to use code you find in the Android or Java APIs.