Description and Plan

Evan Law, Claire Shou, Moussa Zeitoun, Tim Chou

Description:

Group 3's game, "The GPA Challenge," will be a 2D arcade-style game set in SFU's school environment. The player will take control of a student character whose objective will be to collect three to seven grad caps scattered throughout the map to unlock the exit. The player will also be able to collect four randomly generated and limited-time cheat sheets per level, which will raise their GPA (final score) to a maximum of 4.0. As the player proceeds with the primary objective, they will be pursued by rabid and territorial raccoons, instantly ending their playthrough when caught by one. The game will also feature stationary spikes that will lower the player's GPA if the player collides with them. The game will end if the player accrues a GPA lower than zero. After reaching the exit, the game will display the player's final score and time elapsed. Upon completing one playthrough, the player will unlock two additional difficulties, which will determine how many enemies are spawned and the number of grad caps they have to obtain.

Technical tasks:

The board will consist of a 2D string array. It will store entity data in cells, indicating what entity is in each cell. This will allow the program to determine where to spawn bonus rewards and if the player collides with another entity. If the player's score reaches negative after colliding with a trap, they instantly lose. Otherwise, the trap will be despawned for the rest of the level. Moving enemies will utilize an algorithm such as A* search to determine the shortest path to the player. If the shortest route is of length one, it means the player got caught and lost. The game's difficulty presets will be implemented with an enum class using a switch statement. In addition, the group will be using libGDX as our game engine as it is free to use and has a significant amount of documentation and tutorials, making it ideal for new game developers. LibGDX also has a very active and friendly community willing to answer the group's questions should the need arise. The group will be borrowing sprites from the internet for the various entities.

Work Schedule:

The group will meet two to three times per week to coordinate and delegate tasks to members so we can play to our strengths. The group may need to utilize pair programming to ensure members understand the various classes. With these two measures in place, miscommunication between group members will be minimized, and time will be conserved. The group will also need to proactively test new code so we can isolate the cause of the bugs before integrating the latest code with existing code. This will make debugging more manageable as we know which code is defective. The group will also need to experiment with Java libraries, such as graphics and audio libraries.