Architecture Review I

By Yulin Chen, Jeremy Skoler, Alexis Wu

I. Background & Context

We have taken a cautious approach to initiating our final project. We want to make sure we scope the project correctly because we all have negative experiences from poorly scoped projects. This means that although we have decided to create a game, we have not yet settled on the specifics of the game play. We have, however, decided several of the traits our game must have in order for us to be satisfied with the final product. They are listed below:

- Multiplayer mode - From mini project 4, we all came to appreciate the inherent fun in competing with our peers. As we want the game to be an enjoyable experience we have decided it should be a multiplayer game.
- Visually appealing - The visuals of a game enhance the user experience exponentially. We plan to devote a reasonable amount of time to the development of the scenery, character animations and overall visual experience.
- Polished final - As a group we have discussed in length the necessity for a well made, easy to play game that has no bugs. We hope to prioritize this over game complexity so that no matter the actual storyline the game play is smooth and not distracting.

After discussing these requirements we generated a couple game ideas. These include a racing game, a dogfighting plane game and a doodle jump type game. For the purpose of consistency we will focus on the doodle jump game during this review session and make a final decision based on the feedback we receive.

The doodle jump game would involve two players each controlling a sprite. They would be competing to get highest/last the longest. The game would generate platforms around the screen that the sprites could jump to. When the sprites land on a platform there would be a short delay then it would disappear forcing the sprites to keep moving or fall and loose. The multiplayer aspect means that the players sabotage each other by jumping on platforms and preventing their opponent from using it due to its disappearance. We worried about the implementation of gravity in the game so we developed a brief bit of code to test how it would work and found it to be relatively simple.

II. Software Architecture

For the basic model we are considering, the classes for our doodle jump game would be:

- Class Main:
  Incorporates all the classes and updates.

- Class Platform:
  Stores all the sprites that serves as the platform; Detects weather the moving platforms (if applicable) will hit on the players; Updates the platforms as the players hit on it; Determines whether the game is over and who wins.
- Class Level:
  Generates a new line of sprites as new height is reached. The new platforms are stored in the Platform class.
- Class Player1:
  Player 1 controlled sprite. Updates its position and speed (Gravity and parabola calculation included); Detects whether the sprite hits the platforms.
- Class Player2:
  Player 2 controlled sprite. Updates its position and speed (Gravity and parabola calculation included); Detects whether the sprite hits the platforms.
- Class Background:
  Sets the background of the game, might include updating objects like clouds.

III. Key Questions

In addition to tackling technical challenges, our main objective for this review session is to discuss the details in the game design, from a player's point of view, therefore optimizing the user experience in this game play. Potential users' inputs on the following components of the game would be very valuable:

- ❏ Game control (how will the players control the character(s) in the game) - we are currently deciding between the use of OpenCV, gamepad, keyboard for players to control the characters in the game;
- ❏ Story (what would be the best "story" behind the game; who are the character(s)) - it is important for our game to have a "story" that makes sense to the logic and interactions in the game; we currently have various alternatives and would like the users' input on finalizing the theme story.
- ❏ Graphic Design (what is a good visual style that is both visually appealing and makes sense to the story of the game) - we are currently deciding between various visual styles (pixelized, brushwork, minimalistic, etc.)
- ❏ Interactions Between Multiple Players - as we will have a "dual-player mode" for the game, we are deicing the different type of the interactions between the dual battle mode to make the game more fun and exciting: 1) shooting bullets between players; 2) the platform will disappear once one player have touched it, etc.

Thus, to collect suggestions and input for the decision-making in the above-mentioned components, we have the following key questions to guide our review session:

- What has been your favorite game control? Which method of control do you think is best suitable for this particular game?
- Between our various alternatives, what would be a cool story that makes the most sense to the game?
- What artistic (visual) style do you think is the best (most visually appealing)?
- What do you think would be the most exciting ways to compete (with another player) in this game setting?

IV. Agenda for Technical Review

- Background & Context Overview - 1 minute
- Demonstrating Prototype, UX Designs, Visuals - 2 minutes
- Software Architecture - 2 minutes
- Risk Identification and mitigation - 2 minutes
- Feedback - 3 minutes

V. Prototypes

Please view our prototype & presentation slides here