

Individual Pitch:

Team Member: Kate Zraly

One-liner: As the lead UI developer, I will create the Layout, Button and Textbox elements, and implement the start/game over screens.

Outline: Each Layout represents a group of items on the screen, similarly to an HTML <div>. The two main layouts will include the start screen and the game over screen. The game will switch between the start screen, the gameplay, and the game over screen through game states.

Technical Details: The layout will be a composition of buttons, textboxes, and other layouts. When the user clicks on the screen, the Window notifies the input handler. Input handler iterates through the Game's buttons ArrayList and checks to see if the mouse is in each button's bounds, and if so, executes it.

We will include a ScoreBox and PlayButton that both inherit from Textbox and Button. I will handle the game state in Game's update method, so that a certain UI will be rendered for each game state.

Impact: While not part of the core gameplay, I will be enhancing the user's experience by introducing them to the title of the game on the start screen, and showing their final score on the game over screen. The game over screen will have a replay button which will allow the user to play as many times as they wish. The user can also see their score as they play the game.

Timelines:

1 week - Creating the skeletons for Layout, Button, and Textbox.

1 week - Implement Layout's composition quality.

2 weeks - Since I have not yet worked with Java processing, it should take me a week or two to get the Textbox and Buttons on the screen. This also involves creating class variables and the designated setters and getters.

1 week - Implement score.

1 week - Button execution.

2 weeks - Implementing the game over and starting screens.

1 week - Final touch ups and cleaning up the code.

Risks and Mitigations:

I have little experience working with Java processing and understand little about drawing on the screen. To not let this become an obstacle in my work, I will read the Java processing documentation in-depth to gain a solid foundation.