COSC-112-S18
Annabelle Gary, Sylvia Frank, Amy Pass
Final Project Updates

a-MAZE-ing Maze!!

DESIGN CHANGES

HIGH LEVEL DESCRIPTION

- Originally, we had planned to implement an obstacle in which the user had to jump over a gap in the path. However, because we decided to create the game from an aerial perspective, implementing jumps was difficult visually. As a result, we did decided to remove gaps in the path from the list of possible obstacles.
- We added a feature in which points increase when the user kills aliens and when the user collects coins. These features make it easier for the user to increase their points.
- Instead of using the arrows to navigate the marble, as originally planned, the user now uses the ‘i’, ‘l’, ‘k’ , and ‘j’ keys to move up, right, down, and left, respectively, instead of using the arrow keys. We changed the input to better user experience because on some computers, the arrow keys are too small to use comfortably. The user also uses the ‘a’ key to shoot ammo and ‘h’ key to open and close the help menu. We added these functionalities to our original plan because we hadn’t thought to include user-released ammo or a help menu.

DETAILED DESCRIPTION

- Originally, we planned to have the user set a difficult level in the start menu. Instead, we created a class named TriggerEvents that generates a new event (game speed boost, an alien army, or a lot of coins) every time the user gains an additional 25 points. Now, the game becomes more difficult over time because the probability of TriggerEvents generating a new events that makes the game more difficult (game speed boost or an alien army) is high. As a result, the game is responsive to the user’s abilities rather than having the user evaluate their own abilities at the beginning of the game.
- We originally thought to permute segments of paths onto the screen randomly based on what the previous path was. However, we later determined that it would be more efficient to create pre-set LinkedLists that contained a particular configuration of paths. Each set of paths starts and ends in the same place, allowing the program to place them in any given order. These LinkedLists are arranged in a random order to make the game unpredictable and exciting.
- We added a subclass of Path named Horizontal because requiring the user to navigate the marble horizontally, more than a corner would require, while the path continues to move down (making it appear as if the marble is moving up) makes the game more difficult.
- We added another subclass of Path named BigRect. BigRect is a large rectangle that is added to the very beginning of each game. The BigRect space allows the user to play around with the marble’s movement before entering the narrow portion of the path. We

added this feature because it allows the user to experiment with the marble's movement before the game becomes more difficult.

- We originally thought to include a series of variables in each of the subclasses of Path that defined the length of each edge of the path. For example, TopRightCorner had the variables rightEdgeLength, topLength, leftHorizonLength, and leftVertLength. Now, each instance of Path has three primary variables -- x, enterX, and y -- that define the segment's coordinates. This is more efficient because it stores less variables and makes referencing each segment's location easier.
- To meet one of the project requirements, we added a high score functionality through file input and output. This allows the user to track their progress.
- Instead of having the marble glide across the screen and decelerate after a key is released, the marble now only moves when a key is pressed. We made this change because otherwise, the narrowness of the path made it too difficult and the marble was too apt to fall off the path.

ASPECTS OF ORIGINAL DESIGN PLAN IMPLEMENTED

HIGH LEVEL DESCRIPTION

- We originally planned to create the game from an aerial perspective rather than from the perspective of the marble because the graphics would be easier to code and it would be easier for the user to navigate through upcoming segments of the map.
- We kept most of the obstacles from the original plan (path corners, bombs, and aliens) because they increased the enjoyability and difficulty of the game.
- We kept all of the boosters from the original plan (additional ammo, speed up and slow down, and marble size increase and decrease) because they increased the enjoyability of the game and aid the user when the game becomes more difficult.
- Points increase over time, as originally planned, because that follows the logic of most games. This allows the user to have a goal to reach. The purpose of the game is to beat the previous high score.
- The user still loses if they run off the side of the maze or run into a bomb or alien that has not already been killed. These obstacles make it sufficiently difficult for the user to continue playing.
- The marble will continue to move forward no matter what. The input will simply change the marble's direction (up, right, down, and left). The path will move around the marble to make it appear as if the marble itself were moving. When 'a' is pressed, ammo is released in the direction that the marble is moving at the time of release. When 'h' is pressed, the game pauses and the help menu pops up. When 'h' is pressed again, the help menu disappears and the game resumes.
- The purpose of the program is still to achieve user enjoyment. We want the game to be difficult enough that it doesn't become boring, but not so difficult that it seems impossible.

DETAILED DESCRIPTION

- We kept our original plan to create segments of path using the following subclasses of Path: Straight, TopRightCorner, TopLeftCorner, BottomRightCorner, and BottomLeftCorner. These subclasses allowed us to build the series of LinkedLists that held the possible screen configurations of the path.
- We have a class, Marble, as originally planned, that creates the marble that the user navigates. Within the Marble class, the marble is assigned x and y values and a diameter.
- As originally planned, the items on the path (bombs, aliens, ammo, speed boosters, etc.) are created randomly and some, like speed booster and marble size change, use randomization to generate whether they increase or decrease in speed/size.