

## **Twin Snakes Milestone 2.2**

### *Revised Implementation Plan*

Group Members: Joseph Bess, John Herrick, Laura Shub

- ***Basic Gameplay - Controls, Ball properties, Scoring***

The basic gameplay is similar to how we envision the final game. The mouse input successfully moves the paddle around the screen and successful implementation of the Bullet physics library allows the ball to bounce off the walls and the paddles as desired. Currently pressing space starts the game and moves the ball in the direction of the player and pressing R will reset the ball back to its original position. Our current implementation always has the same force applied to the ball when the game begins, but this will be changed to a random direction in our final game. Scoring is also implemented through collision callbacks between the ball and the forward and back walls. One change that we've made since our original design is making the game board a 3-dimensional cube and allowing the ball to move in all 3 dimensions ignoring the effects of gravity, adding an additional layer of difficulty to the game. One thing we have yet to implement is having the ball speed increase as the round goes on. This will likely be implemented by adding an additional impulse on collision callbacks with the paddles in a direction that corresponds to where on the paddle the ball made contact.

- ***Additional Mechanics - Color changes, opponent "AI"***

In order to make our game have more gameplay depth than simply hitting a ball back and forth, we proposed a "color" mechanic wherein the ball will randomly change color upon collision with one of the paddles, and the next paddle to hit the ball must change its color to match it. We are still planning on implementing this feature, and have given each paddle as well as the ball a "color" value that will be used to keep track of this mechanic. Checking of colors will occur during a collision callback between the ball and the paddles. However, this mechanic will need to be scrapped if we cannot figure out a way to display colors of the ball and the paddles in a satisfactory way, otherwise this mechanic could just cause frustration for the player. Currently we have the paddles and ball set to a specific color for testing purposes to ensure that we can set their colors later on for this mechanic. Pressing K will change the color of the paddle, but we have yet to implement the changed collision detection for this.

Additionally, in order to introduce more of a "competitive" aspect to the game, we have included an enemy paddle for the player to play against. It's base behavior is very simple, as it simply checks the position of the ball at each time step and moves towards it if the ball is sufficiently far away. If time allows, as a reach goal, we might make the speed of the AI paddle scale with the number of points the player has to implement a sort of level system where as you get closer to winning the AI gets more

difficult to beat. This would not be too difficult to implement, however having AI that is essentially unbeatable because it is so fast would be a very frustrating experience, and one that we would need to be careful to avoid.

- ***GUI***

CEGUI has been successfully integrated into our project, and our current demo functionality includes a quit button as well as a scoreboard to display the current number of points held by each player. We plan to add in *start*, *reset*, and *pause* buttons, which should be relatively simple as most of the code for these functionalities has already been written, so it's simply a matter of creating the buttons and subscribing them to the appropriate functions.

- ***Sound***

We have not yet successfully incorporated the SDL library, so our game does not have any sound. This is a priority going forward. Once we have successfully hooked in the SDL libraries to our project, we will be able to have sound effects on collisions, possibly adding a different sound for the ball scoring for the player and the opponent beyond the regular sound of the ball hitting the paddles and the walls.

- ***Bells and Whistles - Particle effects, moving camera***

We currently have some basic particle effects implemented. When the player wins, fireworks appear in the background and while we haven't defined "winning" yet, pressing F toggles the fireworks on and off. When the ball goes out of bounds, it explodes in a collection of colorful particles. We also have the camera move directly behind the paddle for now, keeping both the player paddle and the ball in view at all times. We do have the ability to make the camera movement smoother, however for now we have it tracking the paddle directly.

### ***Control Cheat Sheet:***

- Move the mouse to control the paddle
- Press Space to start the ball moving
- Press K to change the color of the ball
- Press F to toggle fireworks
- Press escape to close window

### ***Running the Project:***

- We included a prebuilt executable in the project named 'assignment2' so just typing './assignment2' while in the root directory should run the executable.