**Capstone Project Planning Document**

**Game Title**

If\_I\_Have\_A\_Robot\_Arm

**Estimated Total Time to Develop**

60 hours

**Intermediate Object-Oriented Programming for Unity Games Information**

Total Time Spent Developing Wacky Breakout: 40 hours

Estimated Time or Actual Time? Estimated

Followed Detailed Instructions (Yes or No)? Yes

**If\_I\_Have\_A\_Robot\_Arm Requirement Specification**

If\_I\_Have\_a\_Robot\_Arm is a 3D game for the player to explore and understand the motions of a robotic arm. It is not a game for entertainment purposes but for research and development of robotic systems. I will still call it a game instead of a robot simulator (the two are more synonymous than you think).

The objective is to use the robotic arm in the game to push the cubes and balls (to-be-delivered package) from random locations on the plane to a designated location (target).

Robotic arm:

1. Robotic arm object stem from turret base that fixed relative to the plane/game world
2. Each segment of the arm consists of at least a cylinder
3. The segments are connected by the shoulder, elbow, and wrist joints (does not have to be visually accurate)
4. The movement of each joint is limited to only one axis and the range is bounded. In other word each segment of the arm can only rotate within 360 degrees and cannot go through another segment.
5. The arm rotates by torque on the joint. Each segment is affected by gravity and by the adjacent joints. Note that this does not have to be physically accurate. It does need to allow enough generality to use physics simulator plugin in future iterations.

To-be-delivered package:

1. Cubes: stay still unless forces act on it.
2. Balls: can roll after force is applied.
3. Weight and friction coefficient can be adjusted through game difficulty.

The robotic arm, package, and game plane can interact with each other through collision, friction, and normal force.

The object will disappear when it reaches the target, and points will be added to the score board. When all the objects are moved to the target the player win.

Menus:

1. Main Menu: start game, controls, settings, quit. Menu also display previous win time.
2. Controls: shows the hotkeys for moving each joint
3. Settings: enable adjustment of difficulties by changing package weight/friction and robotic joint torque.
4. Pause Menu: same as Main Menu except that the start button become the continue button

Return to the Main Menu when the game ends. Game time is recorded after each win. It is displayed in the main menu and stored in a separate file. The settings are also stored in a separate file and adjustable by developers. The previous win time and configurations values are passed into every new game.

We will not implement sound for this game.

**Wacky Pong Requirements Specification**

Wacky Pong works a lot like regular Pong, where letting a ball go by scores a point for the other player. The game, however, also spawns a new ball every few seconds, so there are regularly multiple balls in play at the same time.

In addition, there are several different ball types:

1. Traditional. Behaves in the normal way
2. Bonus. Counts for double hits when hit by paddle. Scores double points when missed
3. Freezer. Makes the opponent paddle unmovable for a short period of time. This is a pickup, so it’s destroyed when a paddle hits it
4. Speedup. Makes all balls in play move at twice their current speed for a short period of time. This is a pickup, so it’s destroyed when a paddle hits it

Each ball and pickup only stays alive for a certain period of time. When the time expires, the ball is destroyed.

The game keeps track of and displays each player’s score and the number of times each player has hit a ball with their paddle.

The game ends when the first player scores a certain number of points. At that point, the game displays a message indicating which player won. When one of the players chooses to close the message, the game returns to the main menu.

If a player clicks the play button on the main menu, the game lets them select a difficulty level (Easy, Medium, or Hard), then starts a game with both players playing at the selected difficulty level.

The game has the following menus:

1. Main Menu: Lets a player pick play, help, or quit
2. Difficulty Menu: Lets a player pick Easy, Medium, or Hard for the game
3. Help Menu: A single page that displays brief game instructions
4. Pause Menu: A menu displayed if a player pauses a game in progress. Provides options to resume the game or quit to the main menu

The game stores important game configuration information in a file it reads in at runtime. This approach supports tuning patches later as necessary.

During gameplay, sound effects are used to indicate when a ball has been spawned and when a ball has collided with the paddle or another ball. Gameplay sound effects also indicate when the freezer and speedup effects have been activated and deactivated. Finally, appropriate sounds are played when a ball is lost and the game is lost. The only menu sound effect is a click when a menu button is clicked.