

Society Ball

Authors: Daniel Huber, Richard Zhang, Ryan Tsai

Revision: 5/23/21

Introduction:

In a few paragraphs totaling about ½ page, introduce the high-level concept of your program. What this looks like depends a lot on what type of thing you are making. An introduction for an *application* will look different than one for a *game*. In general, your introduction should address questions like these:

What does your program do?

Our program is meant to be played by two players who bounce a ball from one side of a net to the other. Each player can move their paddle in two dimensions to hit the ball. There are many features, like powerups, obstacles on the court, mines, etc, to add interest to the game.

What problem does it solve? Why did you write it?

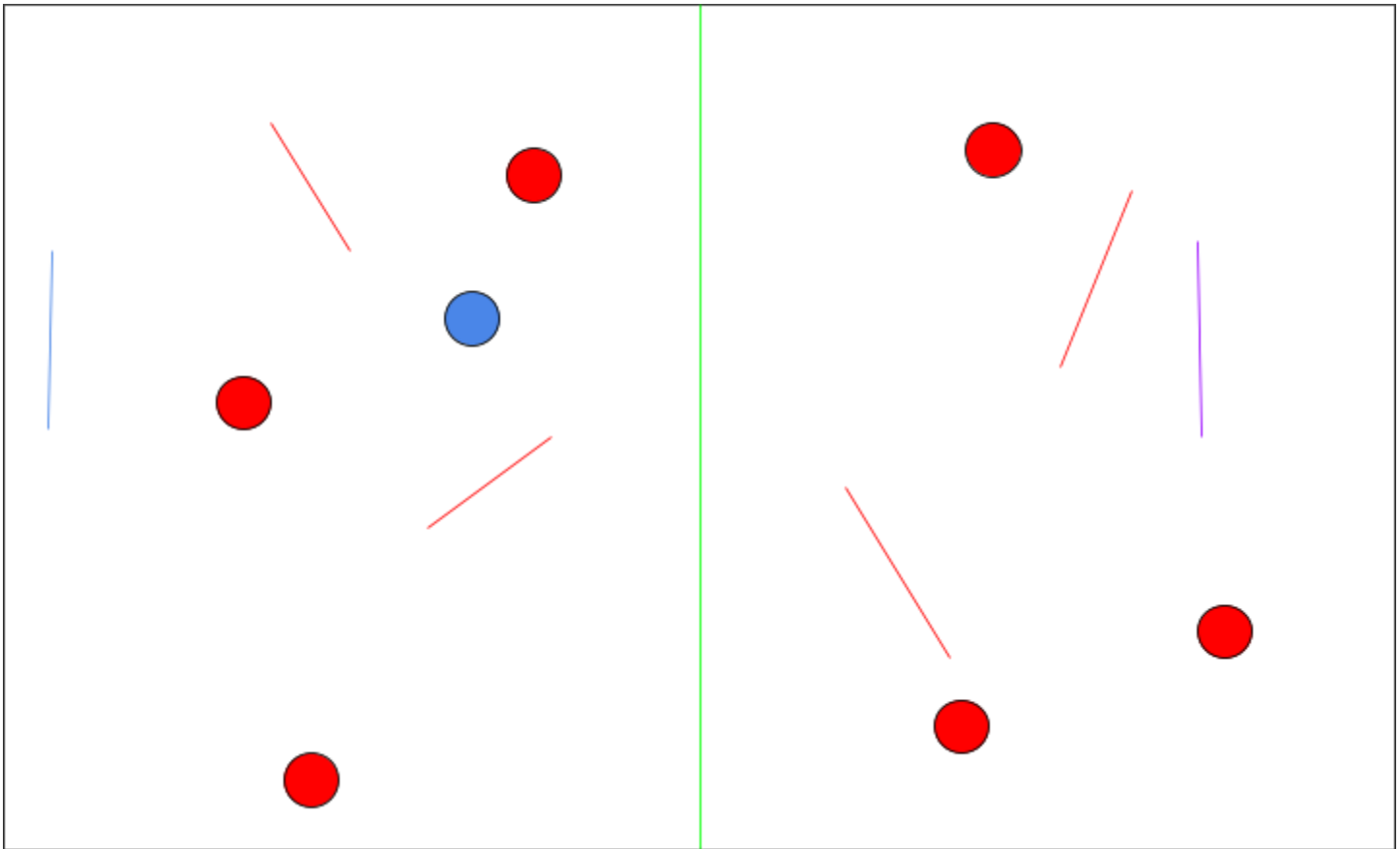
It adds an interesting twist to the well-known game of Pong, but adds many different features that transforms it into a complex and interactive game.

What is the story?

The Hero(blue) and the Devil King(Purple) are breathing their final breaths, with the lives of millions on the line, they set down their bloody swords and decide to settle the war with a game of pong. However, the Devil King used his own Pong board, and set up walls in the hero's side of the board beforehand. However, the hero brought mines and put them on the board to attack the Devil King's paddle. After game one, they decided to make it even and have both walls and mines on both sides, which is when YOU come in as the Hero or the Devil King. There is a divisor, imbued with godly energy that the Hero and the Devil King can not cross. The game is won when the ball hits the edge of the opposite player's wall.

What are the rules? What is the goal?

The goal is simple: hit the ball to the other side such that it hits the opposite short edge of the screen before your opponent can block and hit back with their own paddle. Each time a player does this, they get a point. First player to 10 points wins. The ball will spawn in the middle of the court with a random, but low velocity.



Black rectangle = border of court. When the ball hits the short edge of the court one side, the player on the opposite side gains a point.

Green line = Divider. Players can not cross this line.

Blue lines = players (can move in 2D. Have a sprint function that allows them to move faster, with a cooldown timer)

Blue circle = ball that moves back and forth in 2D.

Red lines = obstacles (the players can not move through these, they must move around them. Balls will bounce off.)

Red circles = mines. If a player hits these they get stuck in place for 2 seconds. In the setup phase before the game starts, each player gets to place three mines in locations of their choosing on the opposite player's side of the court.

Who would want to use your program?

People who want a fun and unique experience playing a Pong-esq game.

What are the primary features of your program?]

Movement mechanics (inspired by pong but does not copy)- 2d, Top down view. Large number of obstacles in the window(Walls, Mines), carrying momentum if sprinting and hitting the ball. Win by having the ball going past your opponent's side. :) For mines, players will input them before the round starts.

Instructions:

[Explain how to use the program. This needs to be **specific**:

Which keyboard keys will do what?

Arrow keys will control movement for the player on the right, WASD will control movement for the player on the left. '.' will control the sprint for the player on the right, 'q' will control the sprint for the player on the left.

Where will you need to click?

You will need to click before the game starts in order to place the mines. You can place your allotted three mines anywhere on the opposing player's side. Player one will place their mines first, then Player two will go.

Will you have menus that need to be navigated? What will they look like?

There will not be menus, but there will be a set-up phase which will require clicking. There will be appropriate instructions for the players. After, the game will start. Please place all mines first before starting the game.

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):

Must-have Features:

[These are features that we agree you will *definitely* have by the project due date. A good final project would have all of these completed. At least 5 are required. Each feature should be fully described (at least a few full sentences for each)]

- The ball will bounce off the paddles
 - The ball will bounce off the paddles with the same speed that it came from. Momentum will be conserved and the angle of the ball's new velocity will be calculated such that it will reflect how a ping pong ball might move in reality (assuming lack of spin and the ball is in a vacuum)
- Movement is based off WASD and Up-Down-Left-Right
 - Each key will make the character move in a different direction. Paddles move only so long as a key is held down. Several keys can be held down consecutively to make the paddle move in more than one direction.
 - For left player:
 - W is up, A is left, S is down, D is right, Left Shift is sprint.
 - For the right player:
 - Up Arrow is up, Left Arrow is left, Down Arrow is down, Right Arrow is right, Right Shift is sprint
- The movement of the ball and paddles will be restricted
 - The paddles will have to avoid obstacles and can not cross the net. If the paddle hits an obstacle, it can not continue moving in the direction, and will stop.
 - The ball will bounce off of the top and bottom walls. If the ball hits one of the side walls, the round will end, as this means the player on the opposing side has won. The ball will bounce off of obstacles as it would off of walls. The ball is not affected by the net or by mines.
- The paddles will be frozen for 2 seconds if they hit the mine(s).
 - Mines, presented in the diagram as small red circles, will act as special obstacles for the players. Before the game starts, each player will be allowed to place three mines on the opposing player's side. The mines will remain there for the rest of the game. Players will

have to take care to avoid mines, because if their paddle touches one, they will be frozen in place for two seconds.

- The paddles will have a sprint capability.
 - When pressing shift for the respective player, the paddle's speed will increase by 50%. Sprint will need to be held down in conjunction with another movement key(s) in order to work. A player will be able to sprint for up to 2 seconds consecutively. There will be an 8 second cool down, during which the player will not be able to sprint at all. The player using sprint will be notified via on-screen alert when the cooldown has gone into effect or has finished. If the player is sprinting and hits the ball, the ball will carry the speed of the sprint.

Want-to-have Features:

[These are features that you would like to have by the project due date, but you're unsure whether you'll hit all of them. A good final project would have perhaps half of these completed. At least 5 are required. Again, fully describe each.]

- Sprinting paddle will carry momentum to the ball
 - If the paddle hits the ball while sprinting, it should also increase the ball's speed by 50%.
- Modifiers
 - Time permitting, we will add the options for players to control master modifiers, including the speed of moving objects, number of points per game, and number of mines and obstacles. For example, a speed modifier of 1.5 would increase the speed of the ball and the paddles by 50% compared to their regular speed.
- Sprint Battery
 - Players can be more specific with their sprint, as they can control how much sprint that they want to use, and it recharges over time, instead of having a hard cooldown for 8 seconds. There will also be a bar in the window which will show the amount of sprint each player has left, and will also show it recharging as time passes.
- Menu
 - This would be a panel of buttons which includes: a way to input modifiers, start game button, choosing which character/side. If you click the modifier button, it will bring up the relevant options to change it.
- Victory Screen
 - Displays the words "The Hero Has Triumphed" or "The Devil King Continues His Reign" depending on which player won. The animation will last for around four seconds, and a new game can be started by clicking with the mouse.

Stretch Features:

[These are features that we agree a *fully complete version of this program would have, but that you probably will not have time to implement*. A good final project does not necessarily need to have any of these completed at all. At least 3 are required. Again, fully describe each.]

- Add Bullets that the paddles can block but if it hits the ball the ball will cause it to go toward the other direction
 - Bullets shoot in a linear motion

- Set speed to be a little faster than the ball
- Wind
 - The wind would be able to be turned on and off with a modifier before the game. If turned on, the wind would then “blow” in a random direction for each point. It would alter the movement of the ball by adding a certain push factor in the direction of the blowing. For example, if the wind were blowing direction North, the ball would move slightly more upwards than expected when in transit between surfaces.
- Terrain
 - Some parts of the court would have inherent modifiers to the movement of the paddles. For example, one part might be asphalt, so a paddle passing over it would have its speed increased by 20%. Another part might be sand, so a paddle passing over it would have its speed decreased by 30%. Terrain would be pre-programmed to be in specific locations on the court, and can be turned on and off through a modifier in the game menu.

Class List:

[This section lists the Java classes that make up the program and very briefly describes what each represents. It's totally fine to put this section in list format and not to use full sentences.]

- Shape
 - Paddle, Ball, Mine, and Obstacles will all extend this class
 - It will contain movement methods, check collisions, color and positions
- Paddle
 - Contains the paddles and controls their movement.
- Ball
 - Contains the ball and controls its movement.
- Mine
 - A mine that will cause the paddle to be stunned for two seconds if it comes in contact with the paddle.
- Obstacles
 - The lines that are drawn in the program, can interact with both Ball and Paddle.
- Main
 - Main method, runs the program.
- DrawingSurface
 - Draws the obstacles and other objects. Also draws court with appropriate dimensions

Credits:

[Gives credit for project components. This includes both internal credit (your group members) and external credit (other people, websites, libraries). To do this:

- List the group members and describe how each member contributed to the completion of the final program. This could be classes written, art assets created, leadership/organizational skills exercises, or other tasks. Initially, this is *how you plan on splitting the work*.

- Give credit to all outside resources used. This includes downloaded images or sounds, external java libraries, parent/tutor/student coding help, etc.]

We all have a focus as specified here, but we all help with all classes:

Daniel - Shape + DrawingSurface + Main + Mine

- Wrote main
- Worked on shape and drawingsurface constructors
- Implemented players placing mines and mine functionality
- Wrote regular paddle movement
- Wrote victory conditions and victory screen

Richard - Shape + Ball + Paddle + Mine + DrawingSurface

- Worked on move methods
- Helped with drawingsurface constructors
- Wrote Sprint

Ryan - Obstacle + DrawingSurface + Paddle

- Completed obstacle/terrain hybrid
- Paddle collisions
- Added a couple methods

Outside resources:

<https://www.youtube.com/watch?v=L6LFmtJynIs>

<https://mkyong.com/java/java-how-to-delay-few-seconds/>

<https://www.youtube.com/watch?v=yKv02lq7JHs>