Benjamin Cohen

Game: Pong Style Game

Features:

* One player-controlled paddle that can go up and down to the edges of the right side of the screen
* A ball that spawns in the middle and automatically starts going towards the player
* ~~On collision with a player or enemy, the ball will rebound at a random angle (between -45 to 45 degrees from paddle normal)~~
* Ball will randomly spawn in a section of the middle of the screen
* Score count will display below player and enemy (first to N points)
* Enemy paddle will go up and down as in a patrol, and not chase after the ball
* Some bitmap image will be in the background

Estimates and Steps

|  |  |  |
| --- | --- | --- |
| 1 | Have ball move back and forth and bounce off walls | 3 hours |
| 2 | Have the bitmap refresh chunks under the ball as it moves | 3 hours |
| 3 | Implement a player paddle and movement | 8 hours |
| 4 | Have bitmap refresh chunks under paddle after move | 3 hours |
| 5 | Implement collision with player | 6 hours |
| 6 | Implement random rebound | 3 hours |
| 7 | Implement enemy + patrol | 6 hours |
| 8 | Implement goal zone where the ball despawns and respawns in middle | 4 hours |
| 9 | Implement score LED sign | 6 hours |
| 10 | Implement score + LED sign show | 3 hours |
|  |  | TOTAL: 45 hours |

Actual Time

Assembly Enemy Patrol + Collision 0:38:13

Assembly Game Ball Move + Bounds Collision 3:52:59

Assembly Game Ball Refactoring 2:14:19

Assembly Game Bitmap Chunk Render 0:33:36

Assembly Game Optimization 8:35:51 (Optimizations + Sounds)

Assembly Game Planning 0:40:08

Assembly Game Player Paddle + Control 3:19:20

Assembly Game Random 1:30:52

Assembly Game Scoreboard 5:25:07

Total 26:50:25

Breakdown:

1. Ball Move
   1. Update every certain amount of timeframe
   2. Takes in an output screen pos and velocity
   3. Has a set size
2. Bitmap refresh
   1. Implemented before or during ball move. Can take in the balls diameter for chunk size and the boundary x,y cords for chunk start and output location
3. Player Paddle
   1. Moves every timeframe if a certain key is being pressed (Up arrow and down arrow)
   2. Has a set size
   3. Takes in x,y position where it starts and adds some value if input is detected
4. Bitmap refresh for paddle
   1. Should be similar implementation to ball move for the paddle (same for score and enemy paddle
5. Collision with player
   1. Additional bounds that needs be detected by the ball move
6. Random Rebound
   1. If the detection is for a player or enemy, then the new velocity will be at some random angle but same velocity magnitude
7. Implement enemy + patrol
   1. Same as player, just movement happens regardless of input
8. Goal zone
   1. Possibly just assign to max bounds since that detection already exists, but reset ball position
9. LED
   1. Score led file will be made to be displayed anywhere
   2. Takes in x,y pos for output, and number to display
   3. Uses a bitmask
10. Score + LED
    1. Have a player and enemy score stored somewhere and those will be reflected by the LED