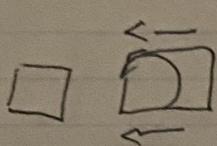
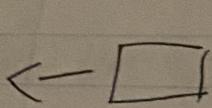


Wall-Box  
if hit reduce health by 1 or  
kill player  
Bounce off & reverse control?



Box-Object  
if hit bounce object back?  
↓ reduce health  
(maybe reset box)



Object  
move forward  
Spawn random Y  
Reset when edge hit

---

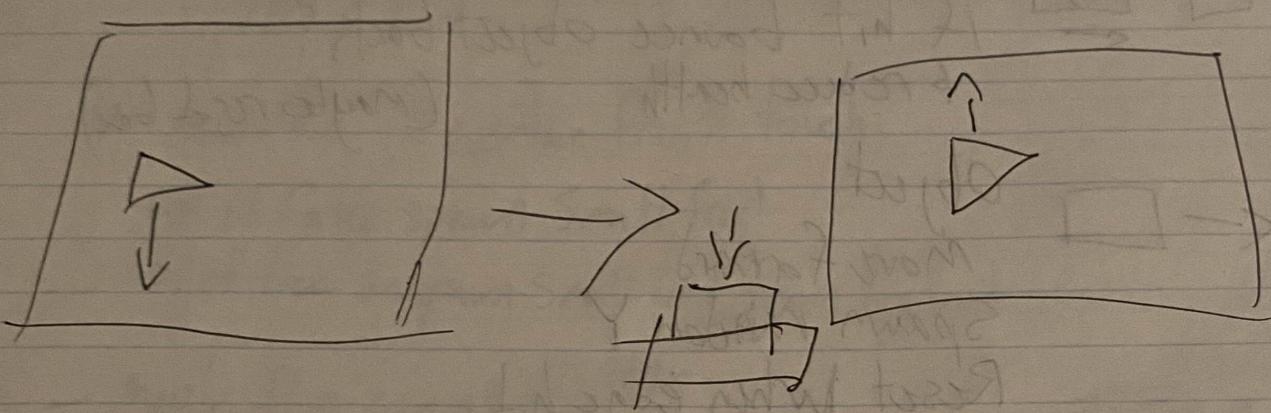
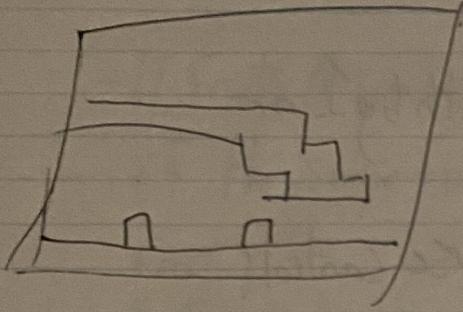
still to do:

→ Plug in Player, Score, audio

Add Collision to main  ~~not yet~~

↳ game ~~miniball~~

→ player has still not been provided. Player/Obstacle collision cannot be added without it. This isn't on Mr. Game over screen has also not provided.



-Character

-Button

-Load images

-Collision (character) / object / wall)

-Audio

-Objects

-Movement

-Score / Game over

```
5 // The namespace your code is in.
6 namespace Game10003
7 {
8     /// <summary>
9     /// Your game code goes inside this class!
10    /// </summary>
11    2 references
12    public class Game
13    {
14        // Place your variables here:
15        Obstacle obstacle;
16        PlayerCollision playerCollision;
17
18        /// <summary>
19        /// Setup runs once before the game loop begins.
20        /// </summary>
21        1 reference
22        public void Setup()
23        {
24            Window.SetSize(600, 600);
25            WindowSetTitle("test");
26            //adds new obstacle class (could turn into an array for extra difficulty?)
27            obstacle = new Obstacle();
28            playerCollision = new PlayerCollision();
29
30        /// <summary>
31        /// Update runs every frame.
32        /// </summary>
33        1 reference
34        public void Update()
35        {
36            Window.ClearBackground(Color.White);
37            //draws obstacle, adds wall collision, and updates position
38            obstacle.DrawObstacle();
39            obstacle.ObstacleWallCollision();
40            obstacle.UpdatePosition();
41
42        }
43    }
44}
45
```

```
2 references
public class PlayerCollision
{
    0 references
    public void Wall()
    {
        //Creates wall collision, commented out currently because i do not have a player class yet
        //Variable names may be incorrect, will plug in player class when created
        // float playerLeftEdge = playerPosition.X;
        // float playerRightEdge = playerPosition.X + playerSize.X;
        // float playerTopEdge = playerPosition.Y;
        // float playerBottomEdge = playerPosition.Y + playerSize.Y;

        //Defines window parameters, right and left can be defined if needed, but are unneccessary for right now
        // bool topOfWindow = playerTopEdge <= 0;
        // bool bottomOfWindow = playerBottomEdge >= Window.Height;

        //if (topOfWindow || bottomOfWindow)
        //{
        //Insert definitions for what happens here (most likely --health or kill screen, that's not on me though)
        //}
    }
}
```

```

{
    Vector2 obstaclePosition = new Vector2(600, 300);
    Vector2 obstacleSize;
    float speed = 300;

    1 reference
    public Obstacle()
    {
        //obstaclePosition.Y = 300;
        obstacleSize = new Vector2(100, 50);
        obstaclePosition.Y = Random.Float(0, 600);
    }

    1 reference
    public void UpdatePosition()
    {
        //updates position
        obstaclePosition.X -= Time.deltaTime * speed;
    }

    1 reference
    public void DrawObstacle()
    {
        //draws obstacle
        Draw.FillColor = Color.Red;
        Draw.Rectangle(obstaclePosition, obstacleSize);
    }

    //obstacle collision, add player class in there \\\\
    0 references
    public void ObstaclePlayerCollision()
    {
        //Import player later
        // float playerLeftEdge = playerPosition.X;
        // float playerRightEdge = playerPosition.X + playerSize.X;
        // float playerTopEdge = playerPosition.Y;
        // float playerBottomEdge = playerPosition.Y + playerSize.Y;

        float obstacleLeftEdge = obstaclePosition.X;
        float obstacleRightEdge = obstaclePosition.X + obstacleSize.X;
        float obstacleTopEdge = obstaclePosition.Y;
        float obstacleBottomEdge = obstaclePosition.Y + obstacleSize.Y;

        // bool doesOverlapLeft = playerLeftEdge < obstacleRightEdge;
        // bool doesOverlapRight = playerRightEdge < obstacleLeftEdge;
        // bool doesOverlapTop = playerTopEdge < obstacleBottomEdge;
        // bool doesOverlapBottom = playerBottomEdge > obstacleTopEdge;

        // bool doesOverlap = doesOverlapLeft && doesOverlapRight && doesOverlapTop && doesOverlapBottom;

        // if (doesOverlap = true)
        {
            //insert --score or kill screen here, not my job though so idk
            Console.WriteLine("ping");
        }
    }

    1 reference
    public void ObstacleWallCollision()
    {
        //resets block once it hits the wall (could potentially track score using this?)
        float obstacleLeftEdge = obstaclePosition.X;

        bool leftOfWindow = obstacleLeftEdge <= 0 - obstacleSize.X;

        if (leftOfWindow == true)
        {
            //resets to a random position once it passes the wall
            obstaclePosition.X = Random.Float(1000, 1500);
            obstaclePosition.Y = Random.Float(0, 600);
        }
    }
}

```