```
        }
        if (Raylib.IsKeyDown(KeyboardKey.Left))
        {
            player2x -= player2Speed;
        }
        if (Raylib.IsKeyDown(KeyboardKey.Right))
        {
            player2x += player2Speed;
        }

        // Draw Player 1
        Draw.FillColor = (Color.Red);
        Draw.Circle(player1x, player1y, 25);

        // Draw Player 2
        Draw.FillColor = (Color.Green);
        Draw.Circle(player2x, player2y, 25);
    }
    0 references
    void PlayerCollisionDetection()
    {

    }
}
}
```

```
            player2x += player2Speed;
        }

        // Draw Player 1
        Draw.FillColor = (Color.Red);
        Draw.Circle(player1x, player1y, 25);

        // Draw Player 2
        Draw.FillColor = (Color.Green);
        Draw.Circle(player2x, player2y, 25);
    }
    0 references
    void PlayerCollisionDetection()
    {
        // Check if Right side of Player 1 collides with Left side of Player 2


        // Check if Left side of Player 1 collides with Right side of Player 2


        // Check if Bottom of Player 1 collides with Top of Player 2


        // Check if Top of Player 1 collides with Bottom of Player 2
    }
}
```

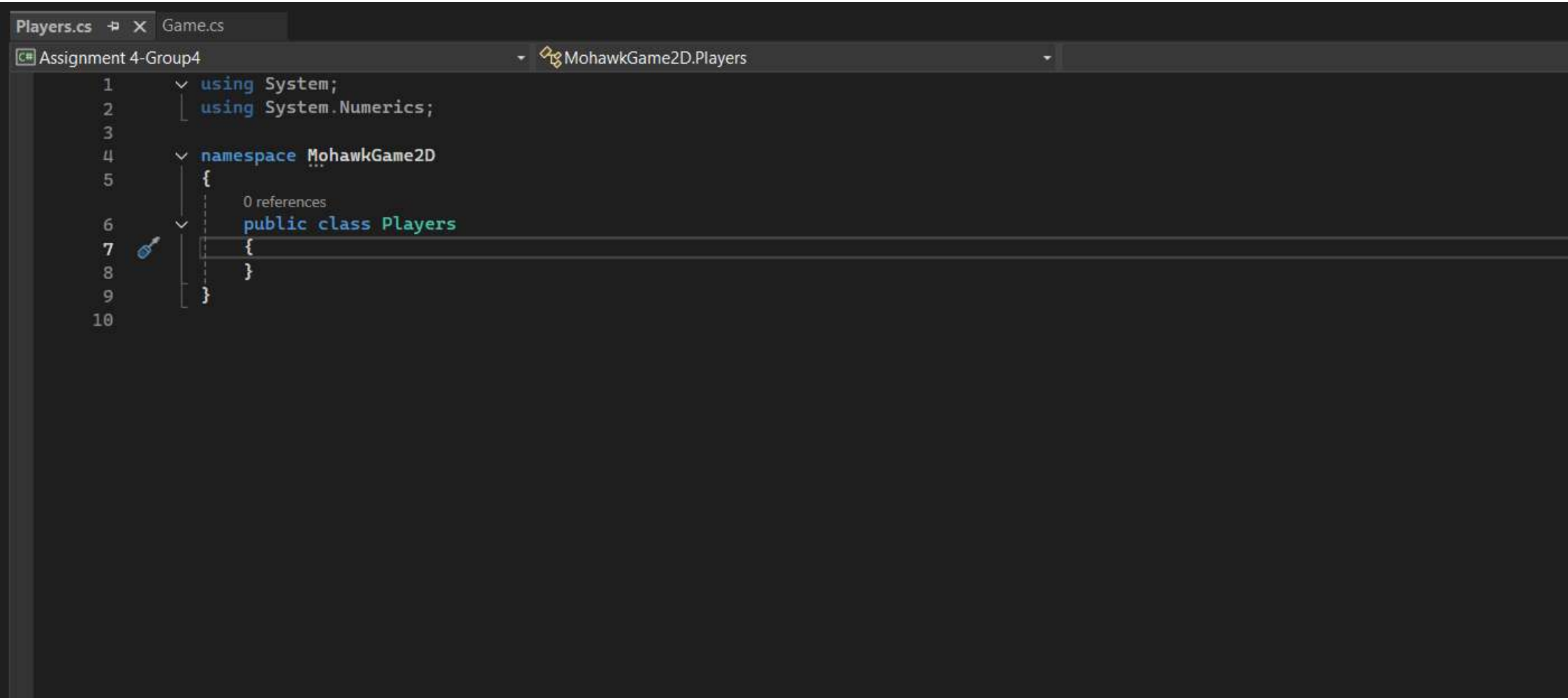
```
        // Draw Player 1
        Draw.FillColor = (Color.Red);
        Draw.Circle(player1x, player1y, 25);

        // Draw Player 2
        Draw.FillColor = (Color.Green);
        Draw.Circle(player2x, player2y, 25);
    }
    0 references
    void PlayerCollisionDetection()
    {
        // Check if Right side of Player 1 collides with Left side of Player 2
        if (player1x + 25 == player2x - 25)
        {

        }

        // Check if Left side of Player 1 collides with Right side of Player 2
        if (player1x - 25 == player2x + 25)
        {

        }

        // Check if Bottom of Player 1 collides with Top of Player 2
        if (player1y + 25 == player2y - 25)
        {

        }

        // Check if Top of Player 1 collides with Bottom of Player 2
        if (player1y - 25 == player2y + 25)
        {

        }
    }
}
```

```
namespace MohawkGame2D
{
    2 references
    public class Game
    {
        // Player 1 XY position
        float player1x = 0.0f;
        float player1y = 0.0f;

        // Player 2 XY position
        float player2x = 0.0f;
        float player2y = 0.0f;

        // Player Speed
        float player1Speed = 4.0f;
        float player2Speed = 4.0f;

        // Player Collision Variables
        bool playerOneLeftCollidePlayerTwoRight = false;
        bool playerOneRightCollidePlayerTwoLeft = false;
        bool playerOneBottomCollidePlayerTwoTop = false;
        bool playerOneTopCollidePlayerTwoBottom = false;
        1 reference
        public void Setup()
        {
            Window.SetTitle("Assignment #  Group #");
```

```
            playerOneTopCollidePlayerTwoBottom = true;
        }
        else
        {
            playerOneTopCollidePlayerTwoBottom = false;
        }

        /*
            SEPERATOR
        */

        // Bounce Players away from each other (P1 = Left | P2 = Right)
        if (playerOneLeftCollidePlayerTwoRight == true)
        {
            player1x -= player1Speed * 2;
            player2x += player2Speed * 2;
        }
        // Bounce Players away from each other (P1 = Right | P2 = Left)
        if (playerOneRightCollidePlayerTwoLeft == true)
        {
            player1x += player1Speed * 2;
            player2x -= player1Speed * 2;
        }
        // Bounce Players away from each other (P1 = Bottom | P2 = Top)
        if (playerOneBottomCollidePlayerTwoTop == true)
        {
            player1y -= player1Speed * 2;
            player2y += player2Speed * 2;
        }
        // Bounce Players away from each other (P1 = Top | P2 = Bottom)
        if (playerOneTopCollidePlayerTwoBottom == true)
        {
            player1y += player1Speed * 2;
            player2y -= player1Speed * 2;
        }
    }
}
```

```
        }

        // Player 2 controls
        if (Raylib.IsKeyDown(KeyboardKey.Up))
        {
            player2y -= player2Speed;
        }
        if (Raylib.IsKeyDown(KeyboardKey.Left))
        {
            player2x -= player2Speed;
        }
        if (Raylib.IsKeyDown(KeyboardKey.Right))
        {
            player2x += player2Speed;
        }

        // Draw Player 1
        Draw.FillColor = (Color.Red);
        Draw.Circle(player1x, player1y, 25);

        // Draw Player 2
        Draw.FillColor = (Color.Green);
        Draw.Circle(player2x, player2y, 25);

        PlayerCollisionDetection();
    }
```

```
        // Draw Player 1
        Draw.FillColor = (Color.Red);
        Draw.Circle(player1x + 100, player1y + 100, 25);

        // Draw Player 2
        Draw.FillColor = (Color.Green);
        Draw.Circle(player2x + 1100, player2y + 100, 25);

        PlayerCollisionDetection();
    }
    1 reference
    void PlayerCollisionDetection()
    {
        // Check if Right side of Player 1 collides with Left side of Player 2
        if (player1x + 25 == player2x - 25)
        {
            playerOneLeftCollidePlayerTwoRight = true;
        }
        else
        {
            playerOneLeftCollidePlayerTwoRight = false;
        }
        // Check if Left side of Player 1 collides with Right side of Player 2
        if (player1x - 25 == player2x + 25)
        {
```


Players.cs
Game.cs
Assignment 4-Group4
MohawkGame2D.Players
using System;
using System.Numerics;
namespace MohawkGame2D
{
0 references
public class Players
{
}
}

```
using System.Numerics;

namespace MohawkGame2D
{
    0 references
    public class Players
    {
        public Vector2 position = new Vector2();
        Vector2 velocity;
        public int size;
    }
}
```

```
using System.Numerics;

namespace MohawkGame2D
{
    0 references
    public class Players
    {
        public Vector2 position = new Vector2();
        Vector2 velocity;
        public int size;
        0 references
        public void Setup(int x, int y)
        {

        }
    }
}
```

PlayerOne.cs | Game.cs

Assignment 4-Group4 | MohawkGame2D.PlayerOne | Setup()

```
4   namespace MohawkGame2D
5   {
        0 references
6       public class PlayerOne
7       {
8           public Vector2 position = new Vector2();
9           Vector2 velocity;
10          public int size;
            0 references
11          public void Setup()
12          {
13              // Setup random XY coords for Player spawn position
14              position = new Vector2(100, 500);
15
16              // Hard code size variable
17              size = 25;
18
19              // Draw Player 1
20              Draw.LineSize = 1;
21              Draw.LineColor = Color.Black;
22              Draw.FillColor = Color.Red;
23              Draw.Circle(position.X, position.Y, size);
24          }
25      }
26  }
27
```

```
            Draw.LineColor = Color.Black;
            Draw.FillColor = Color.Red;
            Draw.Circle(position.X, position.Y, 25);

            PlayerOneControls();
        }
        1 reference
        public void PlayerOneControls()
        {
            // Player 1 Jump
            if (Input.IsKeyboardKeyPressed(KeyboardInput.W))
            {
                position.Y -= 100;
            }
            // Player 1 Move Left
            if (Input.IsKeyboardKeyDown(KeyboardInput.A))
            {
                position.X -= 10;
            }
            // Player 1 Move Right
            if (Input.IsKeyboardKeyDown(KeyboardInput.D))
            {
                position.X += 10;
            }
        }
    }
```

```
            // Player 1 Move Left
            if (Input.IsKeyboardKeyDown(KeyboardInput.A))
            {
                position.X -= 10;
            }
            // Player 1 Move Right
            if (Input.IsKeyboardKeyDown(KeyboardInput.D))
            {
                position.X += 10;
            }
        }
        1 reference
        public void PlayerOneGravity()
        {
            // Calculate Player 1 Gravity
            velocity += new Vector2(0, 40) * Time.DeltaTime;
            position += velocity;

            if (position.Y + 25 > Window.Height)
            {
                position.Y = Window.Height - 25;
                velocity.Y = 0;
            }
        }
    }
}
```

Game.cs | PlayerOne.cs | **PlayerTwo.cs**

Assignment 4-Group4 | MohawkGame2D.PlayerTwo

```
1   using System;
2   using System.Numerics;
3
4   namespace MohawkGame2D
5   {
        0 references
6       public class PlayerTwo
7       {
8       }
9   }
```

```
using System.Numerics;

namespace MohawkGame2D
{
    0 references
    public class PlayerTwo
    {
        public Vector2 position = new Vector2(1100, 500);
        public Vector2 velocity;
        0 references
        public void Setup()
        {
            // Draw Player 2
            Draw.LineSize = 1;
            Draw.LineColor = Color.Black;
            Draw.FillColor = Color.Red;
            Draw.Circle(position.X, position.Y, 25);
        }
    }
}
```

```
            Draw.Circle(position.X, position.Y, 25);

            PlayerTwoControls();
        }
        1 reference
        public void PlayerTwoControls()
        {
            // Player 2 Jump
            if (Input.IsKeyboardKeyPressed(KeyboardInput.Up))
            {
                position.Y -= 100;
            }
            // Player 2 Move Left
            if (Input.IsKeyboardKeyDown(KeyboardInput.Left))
            {
                position.X -= 10;
            }
            // Player 2 Move Right
            if (Input.IsKeyboardKeyDown(KeyboardInput.Right))
            {
                position.X += 10;
            }
        }
    }
}
```

```
        }
        // Player 2 Move Left
        if (Input.IsKeyboardKeyDown(KeyboardInput.Left))
        {
            position.X -= 10;
        }
        // Player 2 Move Right
        if (Input.IsKeyboardKeyDown(KeyboardInput.Right))
        {
            position.X += 10;
        }
    }
    1 reference
    public void PlayerTwoGravity()
    {
        // Calculate Player 2 Gravity
        velocity += new Vector2(0, 40) * Time.DeltaTime;
        position += velocity;

        if (position.Y + 25 > Window.Height)
        {
            position.Y = Window.Height - 25;
            velocity.Y = 0;
        }
    }
```

Game.cs | PlayerOne.cs | PlayerTwo.cs | **PlayerCollision.cs**

Assignment 4-Group4 | MohawkGame2D.PlayerCollision

```
1   using System;
2   using System.Numerics;
3
4   namespace MohawkGame2D
5   {
        0 references
6       public class PlayerCollision
7       {
8       }
9   }
```

```
using System.Numerics;

namespace MohawkGame2D
{
    0 references
    public class PlayerCollision
    {
        0 references
        public void CollisionDetection()
        {

        }
    }
}
```

```
using System.Numerics;

namespace MohawkGame2D
{
    2 references
    public class PlayerCollision
    {
        1 reference
        public void CollisionDetection(PlayerOne playerOne, PlayerTwo playerTwo)
        {
            // Player 1 Sides
            float playerOneLeft = playerOne.position.X - 25;
            float playerOneRight = playerOne.position.X + 25;
            float playerOneTop = playerOne.position.Y - 25;
            float playerOneBottom = playerOne.position.Y + 25;

            // Player 2 Sides
            float playerTwoLeft = playerTwo.position.X - 25;
            float playerTwoRight = playerTwo.position.X + 25;
            float playerTwoTop = playerTwo.position.Y - 25;
            float playerTwoBottom = playerTwo.position.Y + 25;

            // Bounce Players away from each other (Left & Right)
            if (playerOneLeft <= playerTwoRight &&
                playerOneRight >= playerTwoLeft)
            {
                playerOne.velocity.X = -playerOne.velocity.X;
                playerTwo.velocity.X = -playerTwo.velocity.X;
            }
            // Bounce Players away from each other (Top & Bottom)
            if (playerOneTop <= playerTwoBottom &&
                playerOneBottom >= playerTwoTop)
            {
                playerOne.velocity.Y = -playerOne.velocity.Y;
                playerTwo.velocity.Y = -playerTwo.velocity.Y;
            }
        }
```

```
// Transparent Hitbox
Color hitbox = new Color(0, 0, 0, 30);

// Keyboard Input Controls
public KeyboardInput keyJump;
public KeyboardInput keyLeft;
public KeyboardInput keyRight;
2 references
public void Setup()
{
    // Draw Player 1
    Draw.LineSize = 1;
    Draw.LineColor = Color.Black;
    Draw.FillColor = Color.Red;
    Draw.Circle(position.X + 100, position.Y + 500, 25);
    // Player 1 Transparent Hitbox
    Draw.LineSize = 0;
    Draw.FillColor = hitbox;
    Draw.Square(position.X + 75, position.Y + 475, 25 * 2);

    // Draw Player 2
    Draw.LineSize = 1;
    Draw.LineColor = Color.Black;
    Draw.FillColor = Color.Green;
    Draw.Circle(position.X + 1100, position.Y + 500, 25);
    // Player 2 Transparent Hitbox
    Draw.LineSize = 0;
    Draw.FillColor = hitbox;
    Draw.Square(position.X + 1075, position.Y + 475, 25 * 2);

    PlayerControls();
    PlayerGravity();
}
```

Game.cs | Player.cs | PlayerTwo.cs | PlayerCollision.cs | Program.cs

Assignment 4-Group4 | MohawkGame2D.Player | Setup()

```
1   using System;
2   using System.Numerics;
3
4   namespace MohawkGame2D
5   {
        5 references
6       public class Player
7       {
8           public Vector2 position = new Vector2(0, 0);
9           public Vector2 velocity;
10
11          // Keyboard Input Controls
12          public KeyboardInput keyJump;
13          public KeyboardInput keyLeft;
14          public KeyboardInput keyRight;
            2 references
15          public void Setup()
16          {
17              // Draw Player 1
18              Draw.LineSize = 1;
19              Draw.LineColor = Color.Black;
20              Draw.FillColor = Color.Red;
21              Draw.Circle(position.X + 100, position.Y + 500, 25);
22
23              // Draw Player 2
```

```
    }
    1 reference
    public void CollisionDetection()
    {
        // Calculate Player Positions from each other
        float distanceBetweenPlayers = Vector2.Distance(playerOne.position, playerTwo.position);
        float sumOfPlayerRadius = playerOne.size + playerTwo.size;

        if (distanceBetweenPlayers < sumOfPlayerRadius)
        {
            playerOne.velocity.X = 0;
            playerOne.velocity.Y = 0;

            playerTwo.velocity.X = 0;
            playerTwo.velocity.Y = 0;
        }
        else
        {
            playerOne.PlayerControls();
            playerOne.PlayerGravity();

            playerTwo.PlayerControls();
            playerTwo.PlayerGravity();
        }

        //// Calculate Player Positions from each other
```

```
}
1 reference
public void CollisionDetection()
{
    // Calculate Player Positions from each other
    float distanceBetweenPlayers = Vector2.Distance(playerOne.position, playerTwo.position);
    float sumOfPlayerRadius = playerOne.size + playerTwo.size;

    if (distanceBetweenPlayers < sumOfPlayerRadius)
    {
        playerOne.velocity.X = 0;
        playerOne.velocity.Y = 0;
        playerOne.position.X -= 5;
        playerOne.position.Y -= 5;

        playerTwo.velocity.X = 0;
        playerTwo.velocity.Y = 0;
        playerTwo.position.X += 5;
        playerTwo.position.Y += 5;
    }
    else
    {
        playerOne.PlayerControls();
        playerOne.PlayerGravity();

        playerTwo.PlayerControls();
        playerTwo.PlayerGravity();
    }

    //// Calculate Player Positions from each other
    //Vector2 playerOnePosition = playerOne.position;
    //Vector2 playerTwoPosition = playerTwo.position;
    //Vector2 playerOneToPlayerTwo = playerTwoPosition - playerOnePosition;
    //float distanceBetweenPlayers = playerOneToPlayerTwo.Length();

    //// Check if colliding, stops players from moving
    //if (distanceBetweenPlayers < playerOne.size - playerTwo.size)
```