

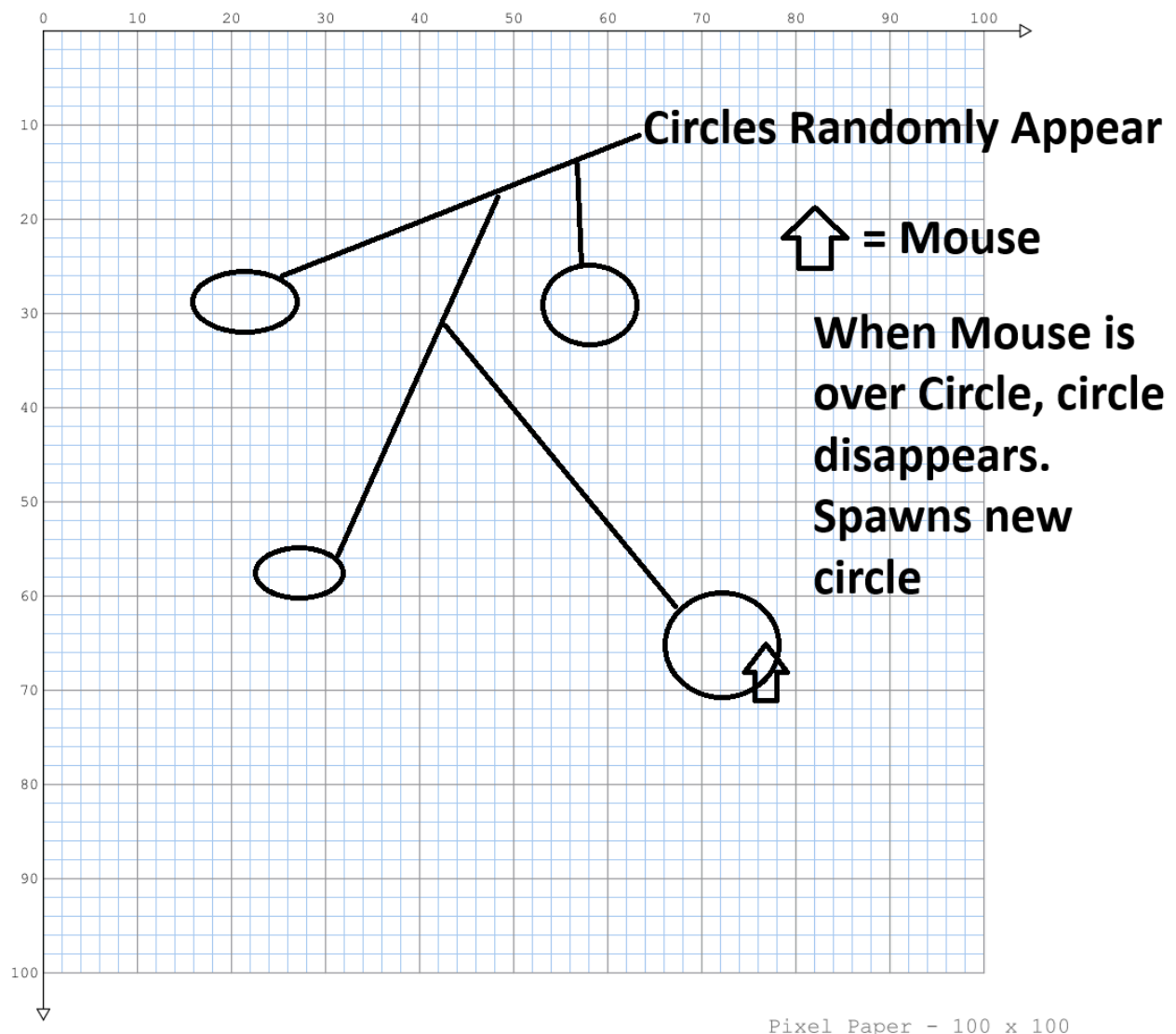
Game Development Assignment 2 Process Notes

I have no fucking idea what I'm doing, so I'm just going to write whatever is in my brain as I have a mental breakdown trying to figure out what I'm doing with my life.

Basically, just spawn random circles, delete circles when the mouse goes over them, and spawn a new circle whenever there isn't already a circle.

Use a boolean to track if a circle exists, use two arrays for randomizing circle spawn locations, one for x one for y.

I don't know how to make the 2d thing detect mouse position, so hopefully I figure that out in the next few hours.



Cannot figure out arrays. They just don't fucking work. Google maybe makes them work??

The screenshot shows a Google search interface with a search bar containing the text: "C# what the actual fuck is this bullshit, I want a fucking Array holding multiple integers, but it just doe". Below the search bar, the "All" tab is selected. The "AI Overview" section provides an explanation: "To make an array holding multiple integers in C#, you must declare it with `int[]` and initialize it with `new int[] { ... }`. The most common mistake is forgetting `new` or treating it like Python." It then provides a "Correct Syntax Example:" in a code block labeled "csharp".

```
csharp
// Method 1: Define and populate immediately
int[] myNumbers = { 10, 20, 30, 40 };
```

Before

```
2 references
public class Game
{
    // Place your variables here:
    bool ActiveCircle = false;
    int VerticalPosition = Array [1, 5, 15];
    int HorizontalPosition = Array.
    int[] myNumbers = { 10, 20, 30, 40 };

    /// <summary>
    ///     Setup runs once before the game loop begins.
}
```

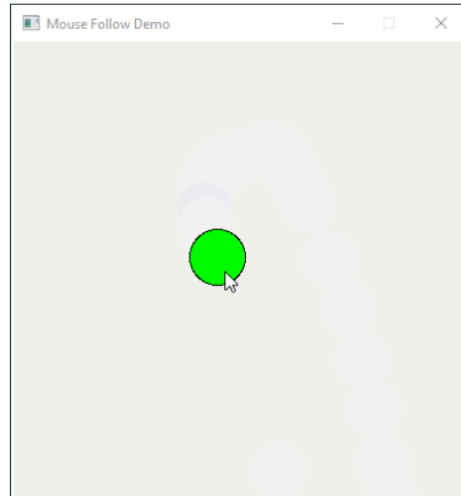
After

```
2 references
public class Game
{
    // Place your variables here:
    bool ActiveCircle = false;
    int[] VerticalCirclePosition = { 10, 20, 30, 40 , 50 , 60 , 70 , 80 , 90 };
    int[] HorizontalCirclePosition = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };
}
```

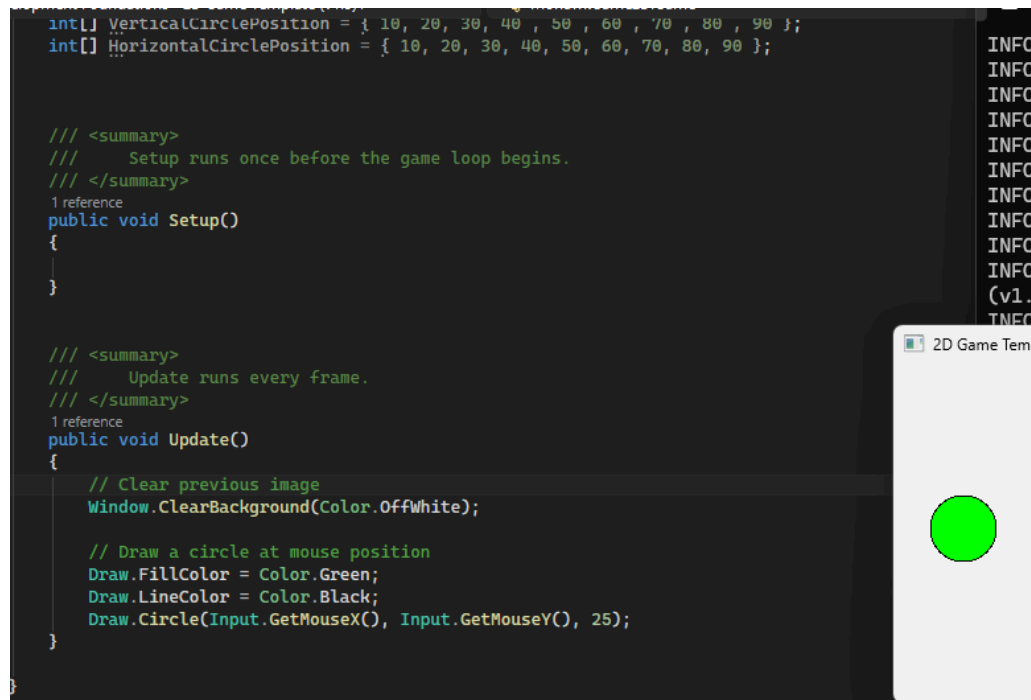
Just gonna rip this code straight into the program because it's 90% the way to step 1 of what I need.

```
public void Update()
{
    // Clear previous image
    Window.ClearBackground(Color.OffWhite);

    // Draw a circle at mouse position
    Draw.FillColor = Color.Green;
    Draw.LineColor = Color.Black;
    Draw.Circle(Input.GetMouseX(), Input.GetMouseY(), 25);
}
```



Thank fuck it's working.



Now here's the hard part, actually drawing the circles, then the harder part of making sure they actually get detected and removed. Probably in a for loop like 50 times or something?

Module 2 - 2D Game Template: Drawing Shapes

2D Game Template: Drawing Shapes

The 2D Game Template provides a special class called `Draw`. `Draw` is used to draw shapes to the screen. There are a wide variety of shapes (`float` values) or using vectors (`Vector2` values).

There are 3 properties used to define the shapes' fill colour, line colour, and line thickness. Set the values at any time. Then, when a shape is drawn, it will use the current values.

```
public void Update()
{
    Draw.FillColor = Color.Red;    // Inside of shape will be red
    Draw.LineColor = Color.Black;  // Lines and outline of shape will be black
    Draw.LineSize = 1;            // Lines and outline will be 1 pixel thick
}
```

Tried to use bits of this and the draw circle on mouse code, ran into errors. Tried to use the randomized number array thing, errors.

It's 1am, I sincerely cannot be fucked figuring this out atm. I'll see if any of the modules have a way of explaining why the actual fuck the int array cant be used here, or what bullshit fucking stupid piece of shit syntax I didn't type correctly so I'm being told to eat shit.

```
18 // For Randomizing Circle Position
19 int[] VerticalCirclePosition = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };
20 int[] HorizontalCirclePosition = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };
21 // For Detecting mouse location
22 float mouseLocationVertical;
23 float mouseLocationHorizontal;
24
25 0 references
26 public void CreateCircle()
27 {
28     if (ActiveCircle == false)
29     {
30         Draw.FillColor = Color.Red; // Inside of shape will be red
31         Draw.LineColor = Color.Black; // Lines and outline of shape will be black
32         Draw.Circle(VerticalCirclePosition, Input.GetMouseY(), 25);
33     }
34 }
35
36
37 /// <summary>
38 ///     Setup runs once before the game loop begins.
39 /// </summary>
40 1 reference
41 public void Setup()
42 {
43     // Clear previous image ( MOVED UP HERE BECAUSE CLEARING BACKGROUND EVERY UPDATE SO
44     Window.ClearBackground(Color.OffWhite);
45 }
```

100% 1 Error 1 Warning 0 of 8 Messages Build + IntelliSense

Error List

Code	Description
CS1955	Non-invocable member 'Game.VerticalCirclePosition' cannot be used like a method.
CS0414	The field 'Game.ActiveCircle' is assigned but its value is never used

????????????????

Gonna just try the same strategy that has been working, copy and pasting module code and altering it slightly.

Module 3 - 2D Game Template: Random Variables

2D Game Template: Random Variables

So far, we have seen that we can be very precise when programming numbers. However, how can we get random numbers?

In games, we often want to produce random (rather, seemingly random) outcomes. Variation in outcomes is often a vital part of a game's design. In the real world, how can we produce randomness? For our needs, we don't need to know how randomness is produced, we simply need to know how to use it.

The 2D Game Template provides a class called `Random`. It has a wide range of functions we can call to produce random values. Let's have a quick look at some list of functions. <https://mohawkraphaelt.github.io/game10003-2d-game-template/>

```
// Get a random Boolean (true or false).
bool coinFlip = Random.Bool();

// Get a random integer in the range 0 through 9.
int integer = Random.Integer(10);

// Get a random float in the range 2f through 4.99999f.
float number = Random.Float(2, 5);

// Get a random color (fully opaque).
Color color = Random.Color();

// There's some other useful stuff in Random, too.
// Like getting a random angle.
float angle = Random.AngleRadians();
```

Another fucking ???? moment

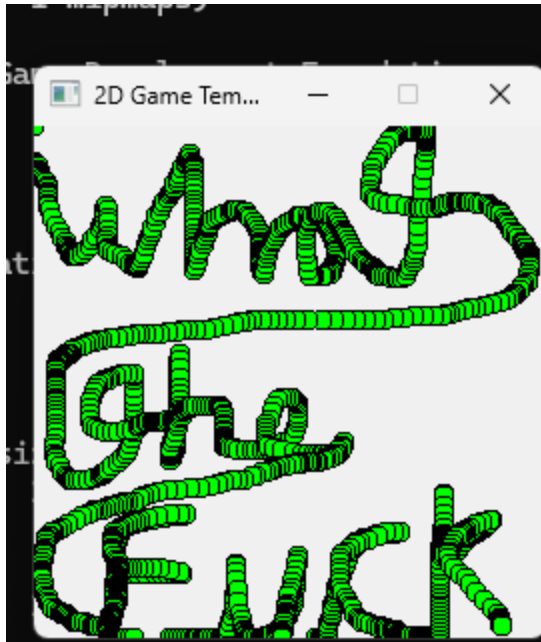
```
/// Update runs every +frame.
/// </summary>
1 reference
public void Update()
{
    // Draw a circle at mouse position ( EXCEPT SMALLER NOW! )
    Draw.FillColor = Color.Green;
    Draw.LineColor = Color.Black;
    Draw.Circle(Input.GetMouseX(), Input.GetMouseY(), 5);

    // This needs to be some kind of check for if the Input.Getmouse X and Y == the circle position

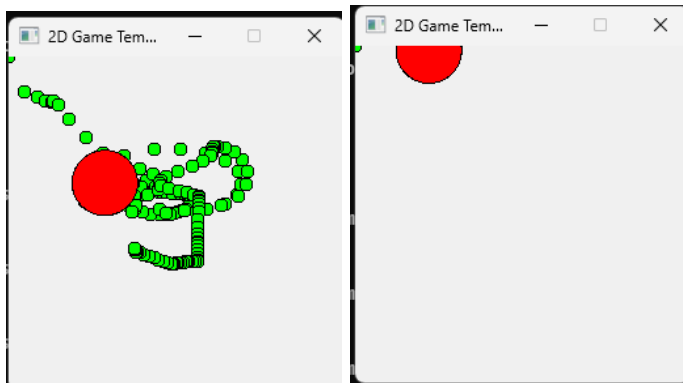
    // Realized I can reuse this to remove the circle as part of the detection/Resetting
    // This SHOULD both clear the screen, and trigger the active circle spawning system.
    if (MouseOverCircle == true)
    {
        Window.Clea (field) bool Game.MouseOverCircle
        ActiveCirc
        CS1003: Syntax error, '(' expected
        Show potential fixes (Alt+Enter or Ctrl+.)
    }
    else
    {
    }

    // Probably also need to spawn the circles under the updates section I imagine
    if (ActiveCircle == false)
    {
    }
}
```

Holy fuck I found it, but why the actual fuck does it not show where the (was expected? I have accidentally Created an Abomination.



GOOD NEWS. Random Circle Spawning is working now.



I LITERALLY DID NOTHING TO THIS, WHY IS THIS CODE SUDDENLY BRICKED?

```

/// <summary>
///     Update runs every frame.
/// </summary>
1 reference
public void Update()
{

    // Draw a circle at mouse position ( EXCEPT SMALLER NOW! )
    Draw.FillColor = Color.Green;
    Draw.LineColor = Color.Black;
    Draw.Circle(Input.GetMouseX(), Input.GetMouseY(), 5);

    // This needs to be some kind of check for if the Input.Getmouse X and Y == the circle
    if (mouseLocationHorizontal)

    // Realized I can reuse this to remove the circle as part of the detection/Resetting

```

LIKE GENUINELY, ARE YOU FUCKING KIDDING ME???

MAYBE ALMOST THERE????

```
// Draw a circle at mouse position ( EXCEPT SMALLER NOW! )
Draw.FillColor = Color.Green;
Draw.LineColor = Color.Black;
Draw.Circle(Input.GetMouseX(), Input.GetMouseY(), 5);

// This needs to be some kind of check for if the Input.Getmouse X and Y == the circle position
if (HorizontalCirclePosition == Input.GetMouseX() && (VerticalCirclePosition) == Input.GetMouseY());
{
    MouseOverCircle = true;
}

// Realized I can reuse this to remove the circle as part of the detection/Resetting
```

But why is the left half giving an error for not being a boolean? I'm not testing for a boolean.

I want to [REMOVED]. Genuinely the error diagnostic shit feels actively malicious in how it seemingly points away from whatever the fuck the problem is meant to be.

```
}

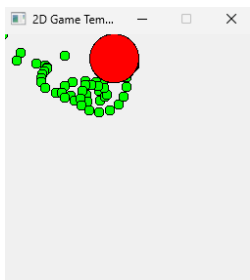
// OK SO I PROBABLY NEED TO MAKE THE MOUSE POSITION BIGGER THAN THE EXACT POSITION. Gonna
int MouseLocation = (Input.GetMousePosition() + 5) & (Input.GetMousePosition() - 5);
```

IM FUCKING LOSING IT

```
// This needs to be some kind of check for if the Input.Getmouse X and Y == the circle position
// GENUINELY WHAT THE FUCK. WHY ISN'T THIS WORKING?!?!?!?!?
if (mouseDetectionX == HorizontalCirclePosition && mouseDetectionY == VerticalCirclePosition)
{
    MouseOverCircle = true;
}

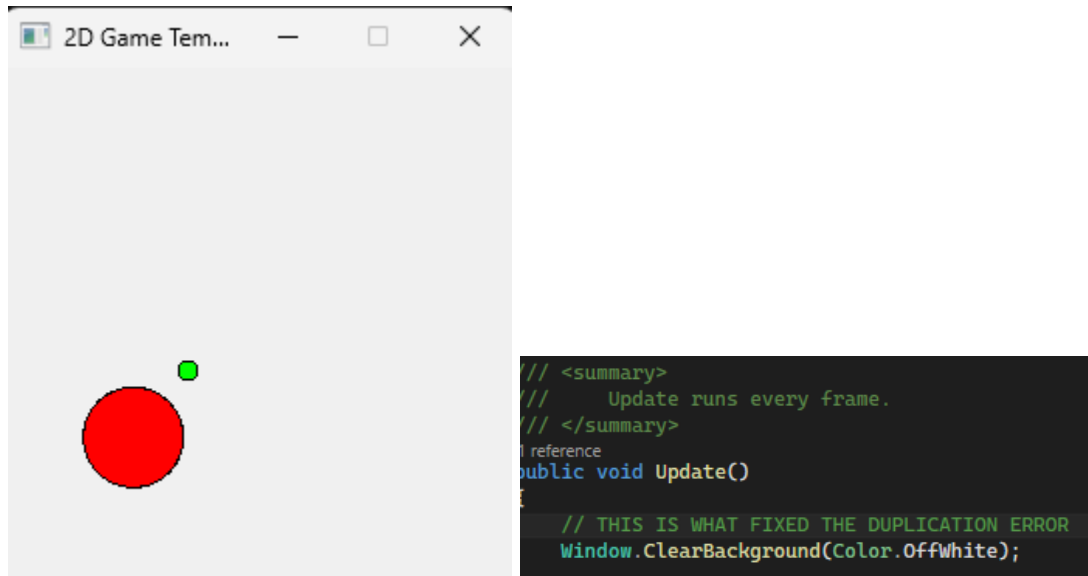
// Realized I can reuse this to remove the circle as part of the detection/Resetting
// This SHOULD both clear the screen, and trigger the active circle spawning system.
if (MouseOverCircle == true)
{
    Window.ClearBackground(Color.OffWhite);
    ActiveCircle = false;
}

// Probably also need to spawn the circles under the updates section I imagine
if (ActiveCircle == false)
{
    CreateCircle();
}
```



I FIXED THE ISSUE, AND ITS JUST NOT DOING IT. Genuinely no fucking idea what is wrong with this atm.

I fixed the duplication error. So either the bug is with the detection of the circle, or the deleting of the circle. I have no idea how to test for which is the actual issue atm, so its time to just throw shit at the wall.



Ok, its intuition, but I think the detection code is the issue. Because that was the hard part, and deletion seems like a very easy matter of shuffling code into the right spot. No fucking idea how to get detection working so more throwing shit at the code wall.

OK OK OK. GOOD NEWS and WEIRD NEWS. Just shoving everything in the updates section seems to be working. Weird news that I cant put in an image, its constantly spawning new circles and deleting the old ones. That means I JUST need to put that code in the detection bit and hopefully it works????

Ok bad news. I think its counting the entire 2d space as the float for the circle location instead of wherever the circle has actually been spawned. So it's just immediately self deleting and then not returning??

No wait, that also makes no sense because then it would be going into seizure mode due to this bit of code at the end

```
// Realized I can reuse this to remove the ci
// This SHOULD both clear the screen, and tri
if (MouseOverCircle == true)
{
    Window.ClearBackground(Color.OffWhite);
    ActiveCircle = false;
}
```

Clearly, the detection isn't working.

I am giving up for now. I'm just going to submit what I have and fix this. It's already 5 days past due. I'll fix this after learning what the actual fuck is wrong with this detection bullshit. I'm struggling with even figuring out how to get the commit into the github repository, relying vaguely on memories from what Owen helped me with during assignment 1.

Apologies if these notes lack professionalism. Were this a group project my use of profanity would be significantly diminished. I have considered cleaning up the language, but I feel the current form these notes take accurately represent my mental state and difficulties with implementation better than more polished language might.