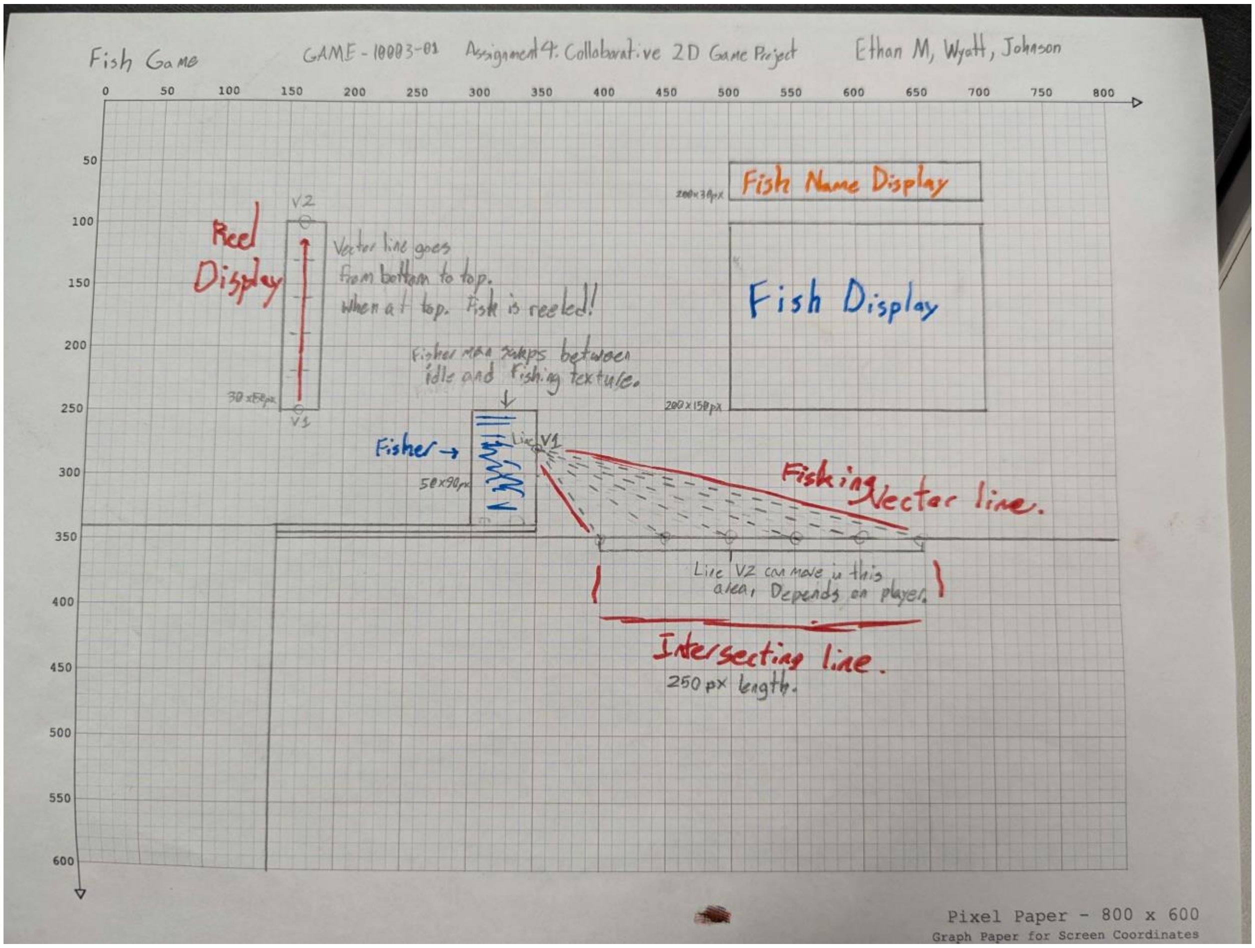


Fish Game GAME-10003-01 Assignment 4: Collaborative 2D Game Project Ethan M, Wyatt, Johnson

Vector line goes from bottom to top. when at top. Fish is reeled!

Fisher man swaps between idle and Fishing texture.

Line V2 can move in this area, Depends on player.

250 px length.

Beginning of Assignment 4 Process Work.

We have agreed to make a fishing game.

The game consists of the player charging their toss, waiting for a fish to bite, then reeling the fish in. Filling a bar to do so.
After the fish is caught it will be displayed in the top right along with its name.

I've created place holder scenery. The next problem is to let the player cast their reel.

I can use Vectors for the line. The second vector will move along the x axis.

For the line:

```
Vector2 Line 1 = new Vector 2 (350,280);
Vector2 Line2 = new Vector2 (lineDistance,350);
line Distance = 400
```

This will be changed by player.

```
    if IsCasting ()
     {
      Draw.LineSize = 2;
      Draw.Line (Line1, Line2)
     }
```

Line Distance caused an error, so I'll simply change the positionX.

To have line go back when not held

```
        if
        ~~while~~ (IsCasting &&!Input.IsMouseButton~~Up~~Down(Left) && Line2 position.X > 400)
         {
          Line2 position.X --1;
         }
```

! Inverts the logic.

To increase the line distance when mouse is held down

```
if (IsCasting && Input.IsMouse ButtonDown (Left) && Line2 <650)
  {
   Line2 position.X ++1;
```

I realize that I want the line to stay still once cast.
What I can do is switch some of my old code to use as code for a marker.
This marker will show up to indicate where the line will be cast.
It will show up while prepping to cast.

(Had to change line 2 positionX to line.2)
(Had to change --/++ to -=/+=

```
if (IsPrepping Cast)
 {
  Draw.Rectangle (Line2.X, 320, 5, 10)
  if (Input. IsMouseButtonDown(Left) && Line2.X<650)
    {
    Line2.X +=1;
    }
  if (!Input. IsMouseButtonDown (Left) && Line2.X>400)
    {
    Line2.X -=1;
    }
 }
```

If I have a "IsIdle" bool I can defierntiate each stage from each other.
I realiced this as I was editing the code to have "IsPrepping cast."
This will help me swap stages without them conflicting with eachother as well!

I'll have four states for now: "IsIdle," "IsPrepping," "IsCasting," "HasCaught".
IsIdle = Idle state IsPrepping = Prepping to cast IsCasting = Fishing HasCaught = displaying fish.

How to swap: (Assume IsIdle = true at start.)

```
if (IsIdle && Input.IsKeyboardKeyPressed (space))
  {
   IsPrepping = true;
   IsIdle = false;
  }
if (IsPrepping && Input.IsKeyboardKeyPressed (space))
  {
  IsCasting = true;
  IsPrepping = false;
  }
if (IsCasting && FishBar =100)
  {
  HasCaught =true;
  IsCasting=false;
  }
```

```
if (HasCaught &&Input.IsKeyboardKeyPressed (space))
  {
  IsIdle = true;
  HasCaught = false;
```

Need to change "IsKeyboardKeyPressed" in some way.
Currently reads all code at the same time.
Temporarily changed code to separate inputs.

suggested solution for game state swapping is to change the bools to ints.

~~if (GameState~~

Possible Code:

```
~~Float~~ int GameState = 0;
    if (GameState <= 4 && Input. Is Keyboard Key Pressed (Key.Space))
      { Float
```

Same Problem

State Machine Transitions (as suggested by prof)

```
int state = 0;

    if (Spacebar Pressed)
      { if (State == 0)
          { State ++;
          }
        else if (state == 1)
          { State ++;
          }
        else if (State == 2)
          { State ++;
          }
        else if (State == 3)
          { State == 0;
          }
```

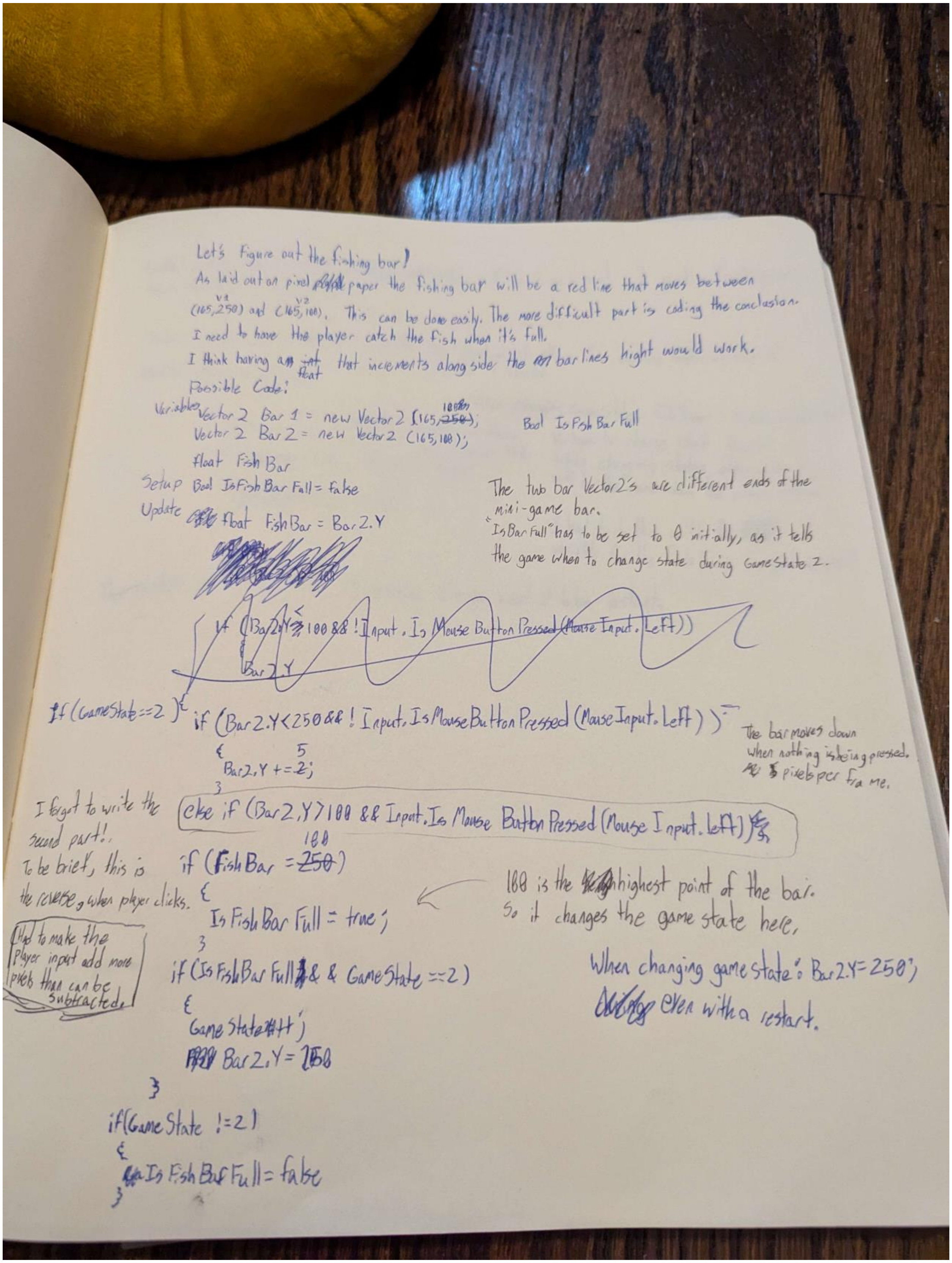
Adding other keys can lead to reseting to first state or others.

```
if (State == 3 && Input. Is Keyboard Key Pressed (Esc))
  { State == 0;
  }
```

```
if (Game State == 3 && Fish Bar = 100)
  { Game State ++;
  }
```

Or:

```
if (Spacebar Pressed)
  { if (gamestate <= 3)
      { gamestate ++;
      }
    else if (Gamestate == 4)
      { GameState = 0;
      }
```



Let's Figure out the fishing bar!

As laid out on pixel paper the fishing bar will be a red line that moves between (165, $\overset{v1}{250}$) and (165, $\overset{v2}{100}$). This can be done easily. The more difficult part is coding the conclusion. I need to have the player catch the fish when it's full.

I think having an ~~int~~ float that incements along side the bar lines hight would work.

Possible Code:

```
Variables  Vector2 Bar1 = new Vector2 (165, 100);
           Vector2 Bar2 = new Vector2 (165,100);

           float FishBar
Setup      Bool IsFishBarFull = false
Update     float FishBar = Bar2.Y
```

Bool IsFishBarFull

The two bar Vector2's are different ends of the mini-game bar.

"IsBarFull" has to be set to 0 initially, as it tells the game when to change state during GameState 2.

~~if (Bar2.Y < 100 && !Input.IsMouseButtonPressed(MouseInput.Left))~~
~~{~~
~~Bar2.Y~~

```
If (GameState==2 ){ if (Bar2.Y<250&& !Input.IsMouseButtonPressed (MouseInput.Left) ) =
                     {
                      Bar2.Y += 5;
                     }
                    else if (Bar2.Y>100 && Input.IsMouseButtonPressed (MouseInput.Left)) {-5}
                     if (FishBar = 100)
                     {
                      IsFishBarFull = true;
                     }
                    if (IsFishBarFull && GameState ==2)
                     {
                     GameState++;
                     Bar2.Y = 250
                 }
            if(GameState !=2)
             {
             IsFishBarFull= false
             }
```

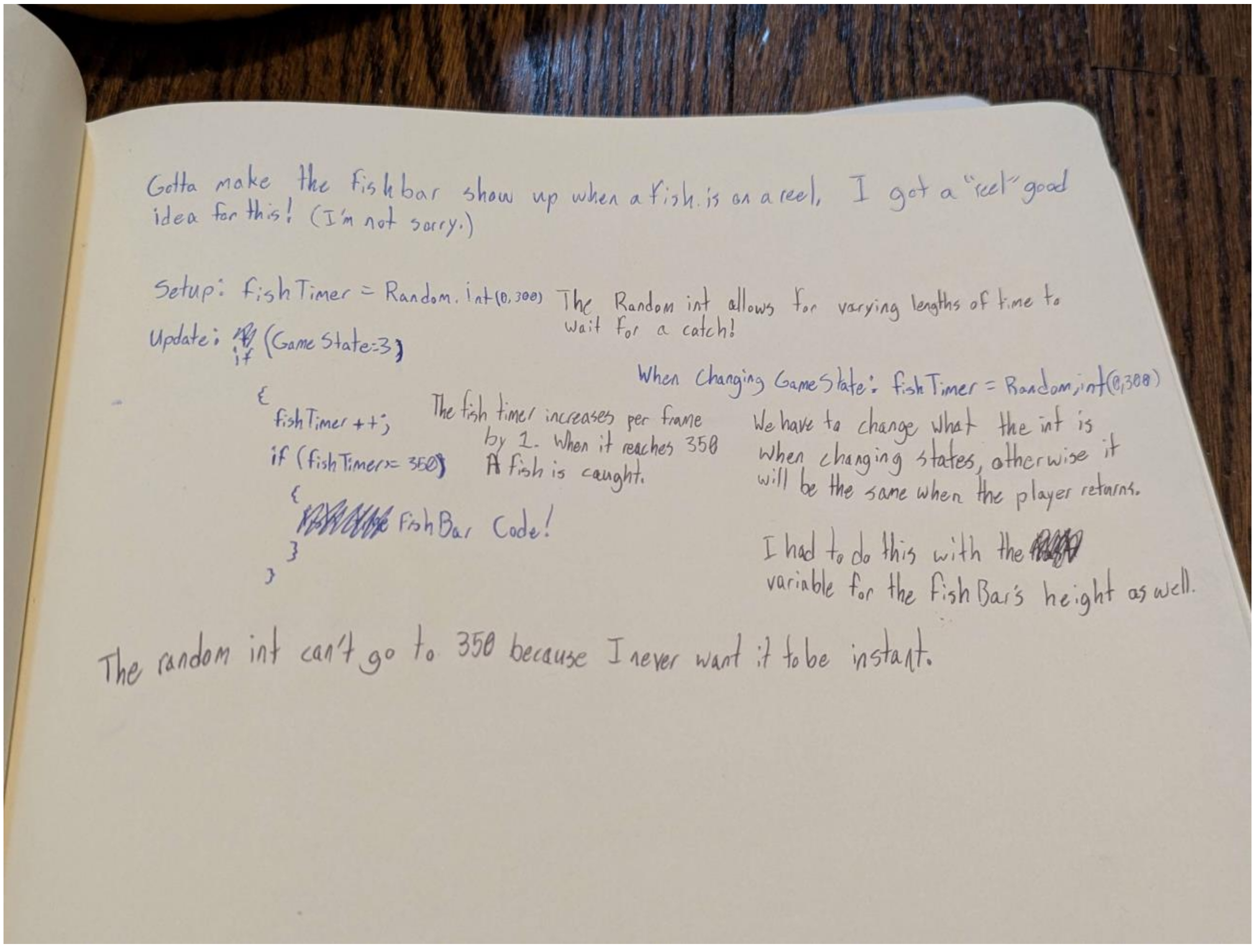
The bar moves down when nothing is being pressed. 5 pixels per frame.

I forgot to write the second part!. To be brief, this is the reverse, when player clicks.

Had to make the player input add more pixels than can be subtracted.

100 is the highest point of the bar. So it changes the game state here.

When changing game state: Bar2.Y=250; even with a restart.



Gotta make the fishbar show up when a fish is on a reel, I got a "reel" good idea for this! (I'm not sorry.)

```
Setup: fishTimer = Random.int(0,300)
Update: if (GameState=3)
        {
         fishTimer ++;
         if (fishTimer>= 350)
          {
           FishBar Code!
          }
        }
```

The Random int allows for varying lengths of time to wait for a catch!

When Changing GameState: fishTimer = Random,int(0,300)

The fish timer increases per frame by 1. When it reaches 350 A fish is caught.

We have to change what the int is when changing states, otherwise it will be the same when the player returns.

I had to do this with the variable for the FishBar's height as well.

The random int can't go to 350 because I never want it to be instant.