

ISSUE LOG/JOURNAL

Issue 1: How do I widen the width of the array of bubbles?

Answer: [Increase the amount of bubbles within the array until it spans the screen](#)

```
Bubble[] bubbles = new Bubble [21];
```

Issue 2: How do I get the spear to move?

Answer: [Following the train example, get the spear to move vertically instead of horizontally by using this](#)

```
void update() {  
    //update spear position  
    spearPosition = spearPosition + spearSpeed;  
  
    if (spearPosition + 50 >= width || spearPosition <= 0) {  
        spearSpeed = spearSpeed * -1;  
        //if time = 18, 16, 14, 12, 10, 8, 6, 4, 2, ySpot = 600  
        ySpot = ySpot * random(-1, 1);  
        //if (ySpot >= height){  
        //ySpot=0;  
        //}  
        println(ySpot);  
    }  
    image(spearPic, spearPosition, -2000+ySpot); //draw spear
```

Issue 3: How do I make a start and restart button?

Answer: [Following the tutorial by Christopher Ockerby on youtube, make a long set of parameters in the constructor function of the button object and map it to left mouse button and the enter key, as well as having it detect the mouse intersection with the button area](#)

```

class Startbutton {
    PVector Pos = new PVector (0, 0);
    float bWidth = 0;
    float bHeight = 0;
    color bColor;
    String bText;
    Boolean bPressed = false;
    Boolean bClicked = false;
    //constructor with parameters to create a start button
    Startbutton(int x, int y, int w, int h, String t, int r, int g, int b)
    {
        Pos.x = x;
        Pos.y = y;
        bWidth = w;
        bHeight = h;
        bColor = color(r, g, b);
        bText = t;
    }
    void update() { //must be placed in draw
        if (mousePressed && mouseButton == LEFT) { //&& bPressed = false
            bPressed = true;
            if (mouseX >= Pos.x && mouseX <= Pos.x + bWidth && mouseY >= Pos.y && mouseY <=
                bClicked = true;
            }
        } else
        {
            bClicked = false;
        }
        if(!mousePressed){
            bPressed = false;
        }
    }
}

```

cting

Issue 4: How do I insert an image to make the player fish move?

Answer: Use PImage as a variable in each if statement for the fish movement

```

}
void move() {
    image(fishyRotate, fishX+sin(frameCount*0.19), fishY);
    // Move the fish
    if (keyPressed) {
        if (key == 'a' || key == 'A') {
            fishX -= fishSpeed;
        } else if (key == 'd' || key == 'D') {
            fishX += fishSpeed;
            image(fishyRotate, fishX+sin(frameCount*0.19), fishY);
            image(fishyRotateright, fishX+sin(frameCount*0.19), fishY);
        } else if (key == 'w' || key == 'W') {
            fishY -= fishSpeed;
            image(fishyRotateup, fishX+sin(frameCount*0.19), fishY);
        } else if (key == 's' || key == 'S') {
            fishY += fishSpeed;
            image(fishyRotatedown, fishX+sin(frameCount*0.19), fishY);
        }
    }
}

```

Issue 5: How do I get the fish to stay on screen?

Answer: Found the system function constrain, and used it with the fish position variables - the size of the image to keep it on screen

```

// Keep the fish in frame
fishX = constrain(fishX, 0, width - 30);
fishY = constrain(fishY, 0, height - 30);
}

```

Issue 6: How do I make a timer?

Answer: Following the tutorial by Christopher Ockerby, using framerate as a way to count down (however, I have been told before that relying on framerate is not a good option so doing something else might work better/more reliably)

```

class Timer {
    float Time;

    Timer(float set) { //constructor
        Time = set;
    }
    float getTime() { // returns the current time
        return(Time);
    }
    void setTime(float set) { //set timer to whatever is temp variable ie. 60 seconds
        Time = set;
    }
    void countUp() { //update time by counting up, called within void draw to work
        Time += 1/frameRate;
    }
    void countDown() { //update time by counting down, called within void draw to work
        Time -= 1/frameRate;
    }
}

```

Issue 7: How do I switch between game states?

Answer: Use switch statements, including the parts of gameplay and conditions for the switches at the end

```

switch(menu) {
case 0://main menu
{
    bigSpear.update();
    fill(250);
    textSize(30);
    textAlign(CENTER, CENTER);
    text("Dodge the spear!", width/2, 300);
    fill(250);
    textSize(30);
    textAlign(CENTER, CENTER);
    text("Swim!", width/2, 350);
    startb.update();
    startb.render();
    if(startb.isClicked()){
        menu = 1;
    }
}
break;
case 1: // game play
{
    startTimer.countDown();
    image(speechB, 230, 360+sin(frameCount*0.3));
    yourFish.move();
    fill(255);
    textSize(42);
    textAlign(CENTER, CENTER);
    text(textPhrases[currentPhrase], width / 2+20, 600);
    if (keyPressed && key == ' ') {
        swapTextPhrase();
    }
    fill(30, 150, 0);
    text(startTimer.getTime(), 1100, 60);
    if (startTimer.getTime() <= 0) {
        menu = 2; // Set menu to "end game"
    }
}
break;
case 2: // end game
{
    gameOver();
    fill(100, 30, 200);
    rect(0, 0, 1200, 700);
    resetbutton.update();
    resetbutton.render();
    textAlign(CENTER, CENTER);
    text("Game Over", 600, 350);
}
}

```

Issue 8: How do I add text for the fish to speak phrases?

Answer: [Make an array of string texts, being able to switch between them by pressing space](#)

```

String[] textPhrases = {"Hey!", "You, yeah you.", "Hello there!", "How's it going?", "Are you fr?"};
int currentPhrase = 0;

speechB = loadImage("speech bobble game sprites.png");

{
    startTimer.countDown();
    image(speechB, 230, 360+sin(frameCount*0.3));
    yourFish.move();
    fill(255);
    textSize(42);
    textAlign(CENTER, CENTER);
    text(textPhrases[currentPhrase], width / 2+20, 600);
    if (keyPressed && key == ' ') {
        swapTextPhrase();
    }
}

```