



# blopaloon

AN INTERACTIVE STORY BY

SYED FAHAD RIZWAN & INSIYA MOTIWALA



# FINAL CONCEPT

## INTERACTIVE STORY

Bloopalloon is an interactive story whereby the user controls a hot air balloon that is constantly deviating from its path because of the wind as it struggles to maneuver past hostile birds that one needs to avoid at all costs as gravity pulls the balloon down.

## MAIN GOAL

Stabilizing the balloon on-screen. Or not. It's your story. Do whatever you want. You cannot win or lose.

## HOW?

Horizontal maneuvering via left and right straps of the "parachute bag gear" which are embedded with force sensors to avoid birds and remain on-screen. To remain vertically afloat, the user has to jump as the bag has an ultrasonic sensor that uses the jump to make a change in the height of the hot air balloon.

## HOW DOES IT END?

It ends after 2 days, regardless of how well you perform. Or if you do not manage to remain afloat for that long and sink below, it ends then too. The difference? Your anonymous help review by the customers you were in charge of in the balloon.

# BRAINSTORMING

## INITIAL IDEAS

While Insiya initially wished to materialize the idea of a plant watering system using a soil moisture sensor, Fahad leaned towards creating an arcade dance machine. Building off from our midterm projects, we realized we wished to detract from "functional, structured, or industrial" projects and create an interactive story. This would be our gift to the world.



## CONTROLLER

Global homogeneity attempts to create a "universal" controller. However, we neither feel organically connected to the experience using a keyboard nor other buttons. We endeavored to create a controlling mechanism in synchrony with the on-screen experience. Hence, we used a bag pack – specifics below.

## ART STYLE

A pixelated retro art style attracted our attention. Insiya used Procreate to create all story elements: backgrounds, birds, and the balloon. Our use of the retro pixelated theme was a homage to the past, but we took a step into the future with our specialized controller.

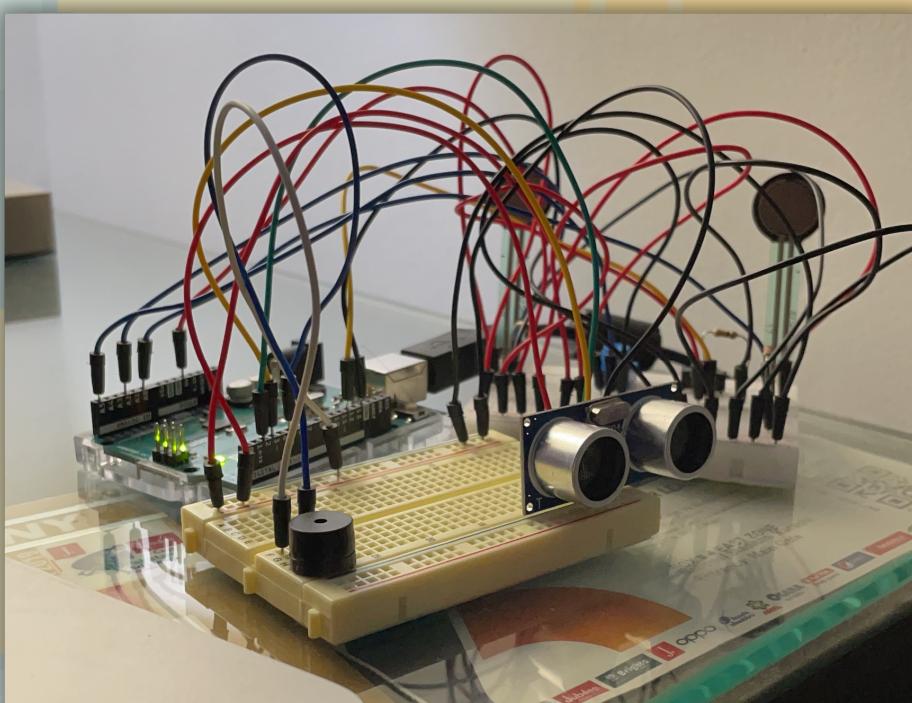
# ARDUINO

## SENSORS & BUTTON

2 force sensors attached to the straps of the bag to maneuver the balloon left and right by mapping the x position of the balloon. Using them, you can avoid the birds that damage your balloon and ensure you remain on the screen.

1 ultrasonic sensor to enable the user to resist the gravity. Once the user jumps, the program calculates the difference in height, compares it with a preset standard and makes the balloon move up by mapping the y position of the balloon.

1 button to begin the story and provide an initial height calibration time.



EXPLORATORY DRAFT

# ARDUINO CODE

Printing inputs on serial monitor to extract them for p5.

```
// printing inputs on serial monitor
Serial.print(forVal);
Serial.print(",");
Serial.print(forVal2);
Serial.print(",");
Serial.print(switchValX);
Serial.print(",");
Serial.println(cleanDistance);
delay(10);
```

Storing all inputs and minimizing noise fluctuations while measuring distance.

```
// storing inputs in variables
forVal = analogRead(A0);
forVal2 = analogRead(A2);
switchVal = digitalRead(12);
switchValX = switchVal + 1;
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
cleanDistance = (cleanDistance * 0.9) + (0.1 * distance);
```

# P5JS CODE

Detecting boundaries to issue warning and increment a counter that influences the ending help review.

```
// boundary detection

// right
if (balloonx > width + 75) {
    yelpCounterPos = 1;
    image(outBound, width - 100, height - 650, 150, 100);
}
// down
if (balloony > height + 75) {
    help();
}
// left
if (balloonx < -75) {
    yelpCounterPos = 1;
    image(outBound, width - 100, height - 650, 150, 100);
}
// up
if (balloony < -75) {
    yelpCounterPos = 1;
    image(outBound, width - 100, height - 650, 150, 100);
}
```

Beginning the story via a button, granting the program sufficient time to calibrate user height which is fixed as soon as they press it.

```
function draw() {
    // start switch condition
    let mode = splitString[2];
    if (mode == 1) {
        image(pixelStart, width / 2, height / 2, width, height);
        textFont(pixelFont);
        textSize(25);
        fill(114, 96, 178)
        if (frameCount %50 <45 ){
            text("PRESS TO START",width/2-120,470)
        }
        baseheight = splitString[3];
        print(splitString[3]);

        // game begins
    } else {
        if (frameCount % 125 == 0) {
            bgcount++;
        }
        if (bgcount > 19) {
            help();
        } else {
            image(sky[bgcount], 400, 360, width, height);
            if (frameCount % 10 == 0) {
                raincount = (raincount + 1) % 3;
            }
        }
    }
}
```

# P5JS CODE

Bird collision detection, health decrement, and counter update that contributes to help review and changes the balloon design to on fire.

```
// bird collision detection
for (let i = 0; i < numbird; i++) {
  distanceX = dist(balloonx, balloonsy, int(birds[i].x),
int(birds[i].y));

  // health decrement upon collision
  if (distanceX < 100) {
    health = health - 2;
  }

  // balloon on fire
  if (health <= 0) {
    g = g - 0.5;

    // yelp bird counter update
    yelpCounterBird = 2;
    // yelp bird counter update
  } else if (health < 100) {
    yelpCounterBird = 1;
  }
}
```

Determining the number of birds on-screen and whether it will rain in any particular storyline inside the setup function.

```
//rain
rain = floor(random(0, 2));
if (rain == 0) {
  r = 0;
} else if (rain == 1) {
  r = r - 1;
}

// birds creation **
numbird = floor(random(2, 5));
for (let i = 0; i < numbird; i++) {
  birds[i] = new bird();
}
```

# P5JS CODE

An extremely detailed and comprehensive chunk of code. It enables balloon control via mapping the balloon's x position using force sensor input.

Calibrates and sets the conditions for the upward movement, connected to the jump by using the distance from ultrasonic sensor. It sets wind via noise, changes balloon (regular and on-fire) using health value, and updates the time counter to contribute to the yelp review.

```
// balloon control input
xposr = map(splitString[1], 0, 550, 0, 10);
xposl = map(splitString[0], 0, 550, 0, 10);

// ultrasonic jump calibration
if (splitString[3] > baseheight + 5) {
  y = y - 2;
}
if (splitString[3] > baseheight + 20) {
  y = y - 3;
} else {
  y = y;
}

// print(y, splitString[3], splitString[1]);
// gravity and noise
g = g - 1;
baloonx = x - n / 3;
baloony = y - g / 2;
xacc = xposr - xposl;
xspeed = xspeed + xacc;
x = 400 + xspeed;
air = air + 0.02;
n = noise(air) * width;

// balloon and burning balloon condition
if (health >= 0) {
  image(pixelBal, baloonx, baloony, 150, 180);
} else {
  image(pixelBalFire, baloonx, baloony, 150, 180);
}

// for yelp review time counter
if (bgcount < 5) {
  yelpCounterTime = 1;
} else if (bgcount < 10) {
  yelpCounterTime = 2;
} else {
  yelpCounterTime = 3;
}
```

# HARDWARE

## BAG CREATION

Since we wanted to create a controller in synchrony with the story concept, our main concern was to place sensors in a way to enable the user to control three analog inputs. Therefore, we chose a bag pack to resemble a "parachute bag," enabling the user to maneuver left and right as well as up and down through the tactical placement of the force sensors and ultrasonic sensor. See images below.

## SOLDERING

In our initial bag creation experiments, it was incredibly challenging to troubleshoot soldered components. However, the multimeter significantly helped us. Additionally, instead of soldering 5V and GND to circuit boards, we braided the wires together to create a knot that was not only rigid and neater but was confirmed to work, or in a rare exception, easy to troubleshoot. However, we soldered all sensors and the button to ensure they remain intact during our story that included a lot of movement.

# BAG PICTURES

BEFORE FABRIC



FABRIC COVERED



ULTRASONIC BELOW



FORCE STRAPS



# SHOWCASE TESTING



# REFLECTIONS

Overall our project was a success! Despite struggling for a long time with certain aspects, we were able to materialize our original vision.

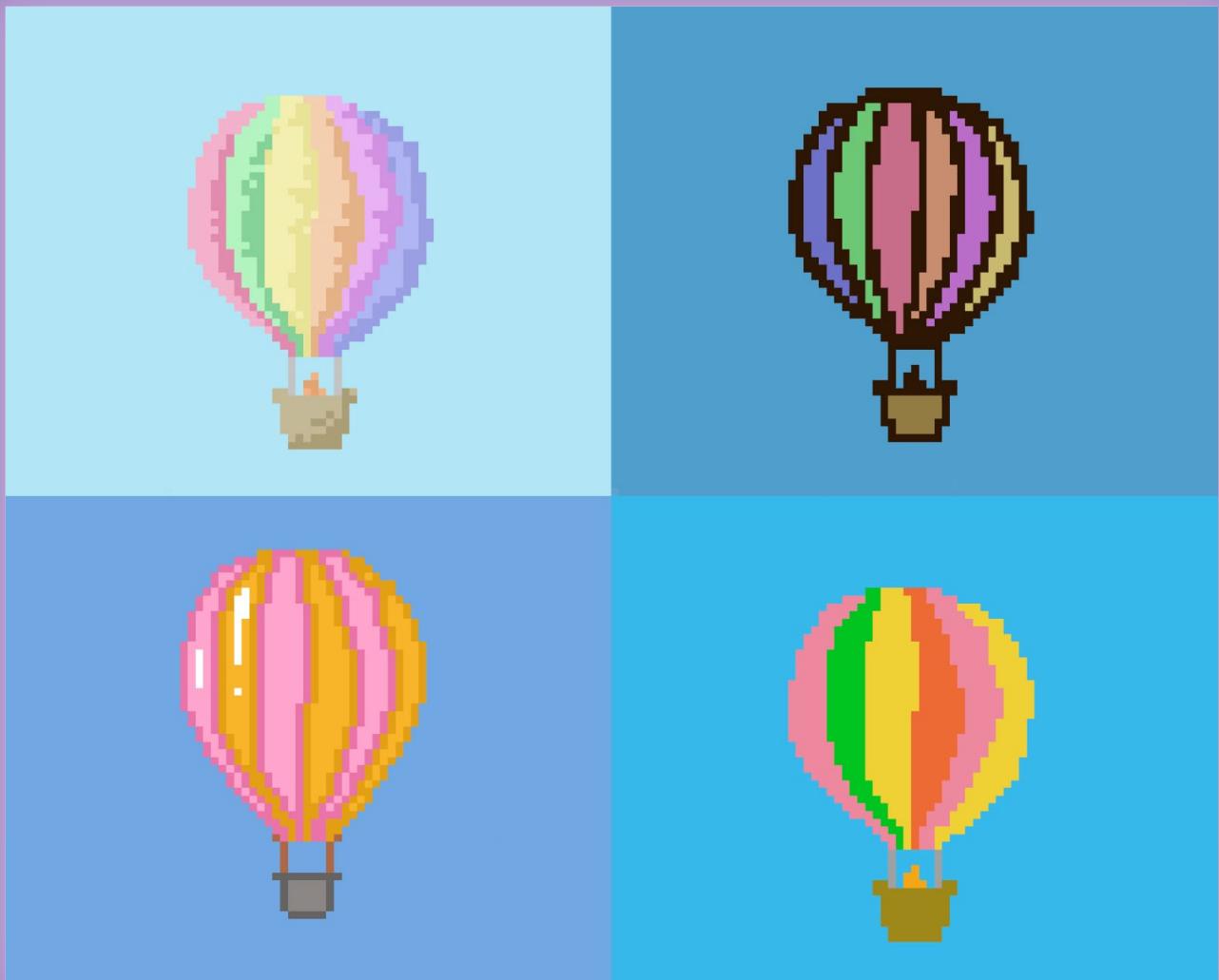
Nonetheless, there are still some improvements we could make in a perfect world. Because we were temporarily using Fahad's school bag, we could not make any permanent changes to it, like cut wholes or sew-on parts, and instead we used removable adhesives like tape to attach everything. This made the entire Project less durable and the tape would peel off and need to be readjusted often. Additionally, we did not add any sounds to the Project. Although this was on our wishlist, we could not find a meaningful way to add sound effects so decided to forgo this extra detail.

For future goals, we both have plans to continue our progress. Insiya will attempt to create a different version of our story using p5.js only so it can be still experienced after the hardware is destroyed. Fahad aims to work on his soldering skills and use his Arduino kit through out the summer to create more projects.

We both really enjoyed this project and the class as a whole and are sad to see it end!

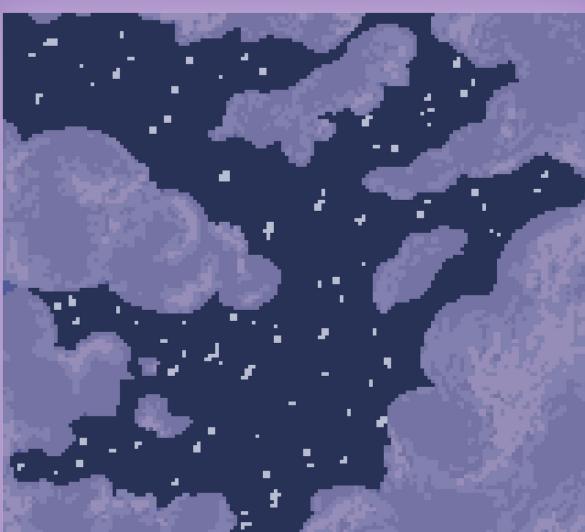
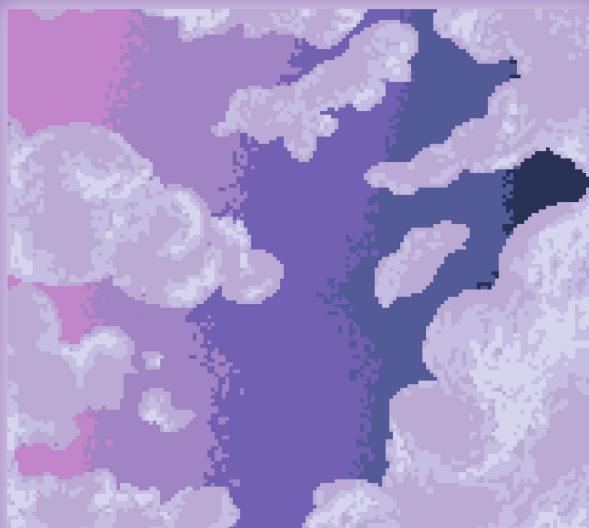
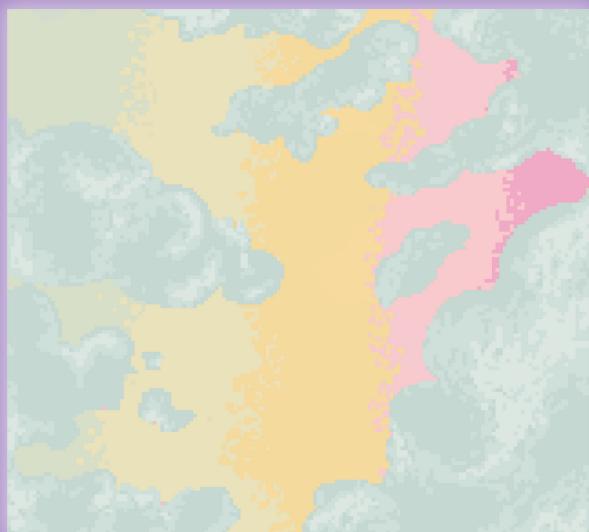
# DIGITAL ARTWORK

Some of the artworks created by Insiya for the Project



This was one of the mock up sketches for the Project as we tried to figure out our art style and what aesthetics we felt would fit best. We decided to settle on the top left one and adjusted it for the final image.

# DIGITAL ARTWORK



# DIGITAL ARTWORK

