116 lines of code per page

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="icon" href=Resources/"ManStand.png" type="image/x-icon">
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <meta name="author" content="YG" />
    <link rel="stylesheet" type="text/css" href="CSS/Menu.css" />
    <!-------Favicon---------------------------------------------------->
    <link rel="icon" href="/Resources/ManStand.png" type="image/x-icon" />
    <title>Path To Freedom</title>
  </head>

  <body>
    <h1>The Path To Freedom</h1>
    <strong><a id="firstlevel" href="Level1/loading.html"> START</a></strong>

    <div></div>
    <a id="Select" href="LevelSelect.html">SELECT LEVEL</a>

    <!----------------------------------- To play the audio ------------------------------------->

    <audio autoplay>
      <source src="Resources/Menu.mp3" type="audio/mp3" />
    </audio>

    <script>
      window.onload = function () {
        let audio = new Audio("Resources/Menu.mp3");

        audio.play();
      };
    </script>

    <!----------------------------------- To play the audio ------------------------------------->

    <video autoplay muted loop id="Menu Video" class="bg-video">
      <source src="Resources/Menu%20Video.mp4" type="video/mp4" />
      Your browser does not support HTML5 video.
    </video>

    <div class="content">
      <!-- Put your content here -->
    </div>
  </body>

  <footer>
    <img id="watermark" src="Resources/Bedroom.png" />
    <p>Made by Chen YG</p>
    <p>Specifically for Ha Khue 2023</p>
    <a id="Developer" href="LevelSelect.html">Developer Mode</a>
  </footer>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-------------------------------------------------------------------------------->

    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Choose your Level</title>
    <link rel="stylesheet" href="CSS/LevelSelect.css" />
    <link rel="icon" href=Resources/ManStand.png type="image/x-icon"></head
  ><!-------------------------------------------------------------------------------->

  <body>
    <!-------------------------------------------------------------------------------->
    <div id="title">
      <h1>Select your Level</h1>
    </div>
```

116 lines of code per page

```
<div class="containers">

    <div class="container1">
      <a href="Level1/loading.html">
          <p>Level 1</p>
          <img src="Resources/Level1preview.png">
      </a>
    </div>


    <div class="container2">
      <a href="Level2/loading.html">
          <p>Level 2</p>
          <img src="Resources/Level2preview.png">

      </a>
    </div>
</div>


  </body>
  <!------------------------------------------------------------------------------>
</html>



<!DOCTYPE html>
<html lang="en">
  <head>
    <!------------------------------------------------------------------>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Password Access</title>
    <link rel="stylesheet" href="CSS/index.css" /></head
  ><!------------------------------------------------------------------>

  <body>
    <!------------------------------------------------------------------>

    <h1>Password Protection</h1>
    <p>For Khue and Friends Only</p>

    <form id="myForm">
      <input type="password" id="myInput" placeholder="Enter Password Here" />
      <input type="submit" value="Submit" />
    </form>

    <script>
      //------------------------------------------------------------------------

      document
        .getElementById("myForm")
        .addEventListener("submit", function (event) {
          event.preventDefault();

          // This prevents the form from being submitted normally-------------------------

          var userInput = document.getElementById("myInput").value;
          if (userInput == "H@1CHAU") {
            window.location.href = "Menu.html";
          } else {
          // do nothing or show an error message-----------------------------------------

            alert("Wrong password, please try again");
          }
        });
    </script>
    <!------------------------------------------------------------------------>
  </body>
  <!------------------------------------------------------------------>
</html>
```

116 lines of code per page

```
body {
    background: black;
    display: flex;
    flex-direction: column;
    align-items: center;
}

*, *::before, *::after {
    box-sizing: border-box;
}


#title {
    color: white;
    font-size: 100%;
    font-family: 'Comic Sans MS', 'Chalkboard SE', 'Comic Neue', sans-seriF;
    text-decoration: italic;
    text-align: center;
}

#title h1 {
    margin-top: 0;  /* Reset margin-top */
}

.containers {
    padding: 3%;
    max-width: 100%;
    display: grid;
    grid-template-columns: 1fr 1fr;;
    column-gap: 3%;
}

.container1,
.container2 {
    text-align: center;
    background: white;
    font-family: 'Comic Sans MS', 'Chalkboard SE', 'Comic Neue', sans-seriF;
    text-decoration: none;
    font-size: 150%;
}

.container1 p,
.container2 p {
    color: black;
}

.container1 img,
.container2 img {
    width: 100%;
    height: auto;
    object-fit: scale-down;
}

a {
    text-decoration: none;
}



body {
    background-color: black;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    text-align: center;
    font-family: comic sans MS, monospace, sans-serif;
}

h1 {
    font-family: comic sans MS, sans-serif;
    color: deeppink;
    font-size: 400%; /* Assuming base font-size to be 16px */
    padding-bottom: 0.0625%;
    margin-bottom: 6.25%;
}

div {
```

116 lines of code per page

```
    padding-top: 3.125%;
}

#Select {
    color: white;
    font-family: comic sans MS, monospace, sans-serif;
    text-decoration: none;
    font-size: 150%; /* Assuming base font-size to be 16px */
    background: darkgreen;
    padding-left: 1.25%;
    padding-right: 1.25%;
    padding-top: 0.625%;
    padding-bottom: 0.625%;
}

p {
    color: white;
    font-family: comic sans MS, monospace, sans-serif;
}

#firstlevel {
    font-family: comic sans MS, monospace, sans-serif;
    color: white;
    background: darkred;
    text-decoration: none;
    font-size: 200%;
    padding-top: 0.625%;
    padding-bottom: 0.625%;
    padding-left: 18.75%;
    padding-right: 18.75%;
    position: relative;
    right: 18.75%;
}

footer {
    margin-top: 10%;
    text-align: center;
}

#watermark {
    background: gray;
    margin: 0.625%;
}

#Developer {
    color: gray;
    text-decoration: none;
    font-family: comic sans MS, arial, monospace, sans-serif;
    font-size: 0.625%; /* Assuming base font-size to be 16px */
}

#MenuAudio {
    background-color: darkblue;
    color: white;
    font-size: 8.125%; /* Assuming base font-size to be 16px */
    text-decoration: none;
    font-family: comic sans MS, monospace, sans-serif;
    padding: 0.1875%;
}

.bg-video {
    position: fixed;
    top: 0;
    left: 0;
    width: auto;
    height: auto;
    z-index: -1;
}

.content {
    position: relative;
    z-index: 1;
    /* Style your content container here */
}


#Back {
  background-color: darkgreen;
```

116 lines of code per page

```
  color: white;
  font-size: 130%;
  width: 14%;
  height: 50px;
  align-items: left;
  margin-left: 1.5%;
  font-family: "Comic Sans MS", "Chalkboard SE", "Comic Neue", sans-serif;
  margin-bottom: 0.9%;
  margin-top: 0.3%;
}

body {
  background-color: black;
  justify-content: center;
}

#Skip {
  background-color: blue;
  position: absolute;
  left: 85%;
  bottom: 90%;
  font-family: "Comic Sans MS", "Chalkboard SE", "Comic Neue", sans-serif;
  color: white;
  width: 10%;
  font-size: 130%;
  padding: 0.5%;
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <!----------------------------------------H E A D---------------------------------------->

    <title>The Path to Freedom</title>
    <link rel="icon" href="../Resources/ManStand.png" type="image/x-icon" />
    <script src="../Libraries/p5.min.js"></script>
    <script src="../Libraries/p5.sound.min.js"></script>
    <script src="variables.js"></script>
    <script src="startgame.js"></script>
    <script src="scene.js"></script>
    <script src="GUI.js"></script>
    <script src="key.js"></script>
    <script src="obstacle.js"></script>
    <script src="animation.js"></script>
    <script src="floor.js"></script>
    <link rel="stylesheet" type="text/css" href="Airport.css" /></head
  ><!----------------------------------------H E A D---------------------------------------->

  <body>
    <!----------------------------------------B O D Y---------------------------------------->

    <a href="../Menu.html"><button id="Back">Back to Menu</button> </a>

    <div id="AirportJS"><script src="Airport.js"></script></div>

    <p id="Skip">Restart Level</p>
    <script>
      document.getElementById("Skip").onclick = function () {
        refreshPage();
      };
      function refreshPage() {
        location.reload();
      }
    </script>
  </body>
  <!----------------------------------------B O D Y---------------------------------------->
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <!----------------------------------------H E A D---------------------------------------->

    <title>The Path to Freedom</title>
```

116 lines of code per page

```
    <link rel="icon" href="../Resources/ManStand.png" type="image/x-icon" />
    <script src="../Libraries/p5.min.js"></script>
    <script src="../Libraries/p5.sound.min.js"></script>
    <script src="variables.js"></script>
    <script src="startgame.js"></script>
    <script src="scene.js"></script>
    <script src="GUI.js"></script>
    <script src="key.js"></script>
    <script src="obstacle.js"></script>
    <script src="animation.js"></script>
    <script src="floor.js"></script>
    <link rel="stylesheet" type="text/css" href="Airport.css" /></head
  ><!----------------------------------------H E A D---------------------------------------->

  <body>
    <!----------------------------------------B O D Y---------------------------------------->

    <a href="../Menu.html"><button id="Back">Back to Menu</button> </a>

    <div id="AirportJS"><script src="Airport.js"></script></div>

    <p id="Skip">Restart Level</p>
    <script>
      document.getElementById("Skip").onclick = function () {
        refreshPage();
      };
      function refreshPage() {
        location.reload();
      }
    </script>
  </body>
  <!----------------------------------------B O D Y---------------------------------------->
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Path to Freedom</title>
    <link rel="icon" href="../Resources/ManStand.png" type="image/x-icon" />

    <script>
        document.addEventListener("DOMContentLoaded", function() {
            document.getElementById("playButton").addEventListener("click", function() {
                var video = document.getElementById("myVideo");
                video.style.display = "block";
                video.play();
            });
        });
    </script>


    <style>
      body {
        background: black;
      }

      h1 {
        background: green;
        text-align: center;
        padding: 1%;
        color: white;
        font-family: comic sans ms;
      }

      a {
        text-decoration: none;
      }

      #playButton {
        font-family: comic sans ms;
        font-size: 150%;
        background: blue;
        color: white;
        width: 50%;
        height: 60px;
      }
```

116 lines of code per page

```
    </style>
  </head>
  <body>
    <!-- This is the button that will trigger the video to appear and play -->
    <button id="playButton">Play Cutscene</button>

    <!-- Initially set the video to be hidden using inline style or CSS -->
    <video
      id="myVideo"
      width="90%"
      height="auto"
      controls
      style="display: none"
    >
      <source src="../Resource2/Loading2.mp4" type="video/mp4" />
      Your browser does not support the video tag.
    </video>

    <div>
      <a href="Airport.html">
        <h1>Continue</h1>
      </a>
    </div>
  </body>
</html>
```

```
body {
    padding: 0;
    margin: 0;
    background: black;
    }

p {
    padding-left: 2%;
    font-family: comic sans MS, sans-serif;
    text-align: left;
    color: white;
    padding-bottom: 0;
    margin-bottom: 0;
    margin-top: 0;
    padding-top: 1%;
  }

h1 {
    text-align: center;
    width: 30%;
    border: solid 1px;
    background-color: blue;
    font-family: comic sans ms;
    color: white;
    display: none;
    cursor: pointer;
}

h1:hover {
    background: yellow;
    color: black;
    cursor: pointer;
}

.info {
    background: black;
    padding-bottom: 1%
}

#defaultCanvas0 {
    border: solid 1px gray;
    margin-left: 15%
}
```

```
//-------DRAW CHARACTERS---------------------//
function drawCharacter() {
  ////////////////////////////////DRAW  CHARACTER//////////////////////////

  //-------------------------FALLING LEFT---------------------------------//
```

```
if (isLeft && isFalling) {
  //_____________________BODY_____________//

  fill(128, 6, 23);
  stroke(0, 0, 0);
  rect(gameChar_x - 6.5, gameChar_y - 50, 13, 30, 4);

  //_____________________HEAD_____________//

  fill(229, 190, 164);
  stroke(0);
  ellipse(gameChar_x, gameChar_y - 60, 25);

  //_________________RIGHT FEET_____________//

  fill(0);
  rect(gameChar_x - 13, gameChar_y - 20, 10, 6);

  //_________________LEFT FEET_____________//

  fill(0);
  rect(gameChar_x + 3, gameChar_y - 15, 10, 6, 4);

  //-----------------EYES----------------//
  fill(0, 0, 0); //left eye
  ellipse(gameChar_x - 6, gameChar_y - 51, 3);
}

//--------------------------JUMP LEFT--------------------------------//
else if (isJump == true && isLeft == true) {
  //_____________________BODY_____________//

  fill(128, 6, 23);
  stroke(0, 0, 0);
  rect(gameChar_x - 6.5, gameChar_y - 50, 13, 30, 4);

  //_____________________HEAD_____________//

  fill(229, 190, 164);
  stroke(0);
  ellipse(gameChar_x, gameChar_y - 60, 25);

  //_________________RIGHT FEET_____________//

  fill(0);
  rect(gameChar_x - 13, gameChar_y - 20, 10, 6, 4);

  //_________________LEFT FEET_____________//

  fill(0);
  rect(gameChar_x + 3, gameChar_y - 15, 10, 6, 4);

  //-----------------EYES----------------//

  fill(0, 0, 0); //left eye
  ellipse(gameChar_x - 6, gameChar_y - 63, 3);
}

//-------------------------JUMP RIGHT-------------------------------//
else if (isRight && isJump) {
  //_____________________BODY_____________//

  fill(128, 6, 23);
  stroke(0, 0, 0);
  rect(gameChar_x - 6.5, gameChar_y - 50, 13, 30, 4);

  //_____________________HEAD_____________//

  fill(229, 190, 164);
  stroke(0);
  ellipse(gameChar_x, gameChar_y - 60, 25);

  //_________________RIGHT FEET___________//

  fill(0);
  rect(gameChar_x + 3, gameChar_y - 20, 10, 6, 4);
```

```
    //_________________LEFT FEET___________//

    fill(0);
    rect(gameChar_x - 13, gameChar_y - 15, 10, 6, 4);

    //---------------------EYES--------------//
    fill(0, 0, 0);
    ellipse(gameChar_x + 6, gameChar_y - 67, 3);
  }

  //----------------FALLING RIGHT-----------------------------------------//
  else if (isRight && isFalling) {
    //_____________________BODY_____________//

    fill(128, 6, 23);
    stroke(0, 0, 0);
    rect(gameChar_x - 6.5, gameChar_y - 50, 13, 30, 4);

    //_____________________HEAD_____________//

    fill(229, 190, 164);
    stroke(0);
    ellipse(gameChar_x, gameChar_y - 60, 25);

    //_________________RIGHT FEET___________//

    fill(0);
    rect(gameChar_x + 3, gameChar_y - 20, 10, 6, 4);

    //_________________LEFT FEET___________//

    fill(0);
    rect(gameChar_x - 13, gameChar_y - 15, 10, 6, 4);

    //---------------------EYES--------------//
    fill(0, 0, 0);
    ellipse(gameChar_x + 6, gameChar_y - 57, 3);
  }

  //----------------------------WALK LEFT--------------------------------//
  else if (isLeft == true) {
    //_____________________BODY_____________//

    fill(128, 6, 23);
    stroke(0, 0, 0);
    rect(gameChar_x - 6.5, gameChar_y - 39, 13, 30, 4);

    //_____________________HEAD_____________//

    fill(229, 190, 164);
    stroke(0);
    ellipse(gameChar_x, gameChar_y - 50, 25);

    //_________________RIGHT FEET___________//

    fill(0);
    rect(gameChar_x, gameChar_y - 10, 10, 6, 4);

    //_________________LEFT FEET___________//

    fill(0);
    rect(gameChar_x - 15, gameChar_y - 10, 10, 6, 4);

    //-----------------EYES----------------//
    fill(0, 0, 0); //left eye
    ellipse(gameChar_x - 6, gameChar_y - 51, 3);
  }

  //---------------------WALK RIGHT-------------------------------------//
  else if (isRight == true) {
    //_____________________BODY_____________//
    fill(128, 6, 23);
    stroke(0, 0, 0);
    rect(gameChar_x - 6.5, gameChar_y - 39, 13, 30, 4);

    //_____________________HEAD_____________//
    fill(229, 190, 164);
    stroke(0);
```

```
    ellipse(gameChar_x, gameChar_y - 50, 25);

    //____________________RIGHT FEET_____________//
    fill(0);
    rect(gameChar_x + 5, gameChar_y - 10, 10, 6, 4);

    //____________________LEFT FEET_____________//
    fill(0);
    rect(gameChar_x - 10, gameChar_y - 10, 10, 6, 4);

    //----------------------EYES--------------//
    fill(0, 0, 0);
    ellipse(gameChar_x + 6, gameChar_y - 51, 3);
  }

  //----------------------------FALLING DOWN-----------------------------//
  else if (isFalling == true) {
    // add your jumping facing forwards code

    //____________________BODY_____________//

    fill(128, 6, 23);
    stroke(0, 0, 0);
    rect(gameChar_x - 6.5, gameChar_y - 50, 13, 30, 4);

    //____________________HEAD_____________//

    fill(229, 190, 164);
    stroke(0);
    ellipse(gameChar_x, gameChar_y - 60, 25);

    //____________________RIGHT FEET_____________//

    fill(0);
    rect(gameChar_x + 3, gameChar_y - 15, 10, 6, 4);

    //____________________LEFT FEET_____________//

    fill(0);
    rect(gameChar_x - 13, gameChar_y - 15, 10, 6, 4);

    //----------------------EYES--------------//
    fill(0, 0, 0); //right eye
    ellipse(gameChar_x + 6, gameChar_y - 53, 3);

    fill(0, 0, 0); //Left eye
    ellipse(gameChar_x - 6, gameChar_y - 53, 3);
  }

  //----------------------JUMP UP-----------------------------------------//
  else if (isJump == true) {
    //____________________BODY_____________//
    fill(128, 6, 23);
    stroke(0, 0, 0);
    rect(gameChar_x - 6.5, gameChar_y - 39, 13, 30, 4);

    //____________________HEAD_____________//
    fill(229, 190, 164);
    stroke(0);
    ellipse(gameChar_x, gameChar_y - 50, 25);

    //____________________RIGHT FEET_____________//
    fill(0);
    rect(gameChar_x + 3, gameChar_y - 10, 10, 6, 3);
    //____________________LEFT FEET_____________//
    fill(0);
    rect(gameChar_x - 13, gameChar_y - 10, 10, 6, 3);

    //----------------------EYES--------------//
    fill(0, 0, 0); //right eye
    ellipse(gameChar_x + 6, gameChar_y - 57, 3);

    fill(0, 0, 0); //left eye
    ellipse(gameChar_x - 6, gameChar_y - 57, 3);
  }

  //-----------------------------STAND FORWARD---------------------------//
  else {
```

116 lines of code per page

```
    //_____________________BODY_____________//
    fill(128, 6, 23);
    stroke(0, 0, 0);
    rect(gameChar_x - 6.5, gameChar_y - 39, 13, 30, 4);

    //_____________________HEAD_____________//
    fill(229, 190, 164);
    stroke(0);
    ellipse(gameChar_x, gameChar_y - 50, 25);

    //___________________RIGHT FEET___________//
    fill(0);
    rect(gameChar_x + 3, gameChar_y - 10, 10, 6, 3);
    //___________________LEFT FEET____________//
    fill(0);
    rect(gameChar_x - 13, gameChar_y - 10, 10, 6, 3);

    //---------------------EYES--------------//
    fill(0, 0, 0); //right eye
    ellipse(gameChar_x + 6, gameChar_y - 51, 3);

    fill(0, 0, 0); //Left eye
    ellipse(gameChar_x - 6, gameChar_y - 51, 3);
  }
}

//------------------------------DRAW FUNCTIONS

//-------------------DRAW STARS----------------------X

function drawStars() {
  let r = 5;

  fill(235, 231, 0);
  ellipse(gameChar_x - 400, 100, r);
  ellipse(gameChar_x + 400, 150, r);
  ellipse(gameChar_x + 400, 150, r);
  ellipse(gameChar_x + 350, 200, r);
  ellipse(gameChar_x + 430, 350, r);
  ellipse(gameChar_x + 300, 150, r);
  ellipse(gameChar_x + 200, 150, r);
  ellipse(gameChar_x + 250, 200, r);
  ellipse(gameChar_x + 170, 350, r);

  ellipse(gameChar_x - 330, 350, r);
  ellipse(gameChar_x - 200, 150, r);
  ellipse(gameChar_x - 100, 150, r);
  ellipse(gameChar_x - 150, 200, r);
  ellipse(gameChar_x - 70, 350, r);

  ellipse(gameChar_x - 400, 150, r);
  ellipse(gameChar_x - 500, 200, r);
  ellipse(gameChar_x - 450, 350, r);
}

function drawClouds() {
  //-------------------CLOUDS ANIMATIONS----------------------X

  for (var i = 0; i < cloud_x.length; i++) {
    gametime = millis();

    if (gametime < 100000) {
      increment -= 0.015;
    } else if (gametime > 100000) {
      increment += 0.015;
    }

    //----------------CLOUD SKETCH----------------------------X

    noStroke();
    fill(200);

    ellipse(
      cloud_x[i].xPos + increment, //make sure that clouds move to the right
      130 + cloudbop, //add bopping effect to cloud
      100,
      100
    );
```

```
          ellipse(cloud_x[i].xPos * cloud_x[i].scale - 50 + increment, 130 + cloudbop, 100, 110);

          ellipse(cloud_x[i].xPos * cloud_x[i].scale - 50 + increment, 100 + cloudbop, 80, 80);

          ellipse(cloud_x[i].xPos * cloud_x[i].scale - 100 + increment, 130 + cloudbop, 100, 100);

          ellipse(cloud_x[i].xPos * cloud_x[i].scale - 50 + increment, 130 + cloudbop, 50, 50);

          ellipse(cloud_x[i].xPos * cloud_x[i].scale - 140 + increment, 130 + cloudbop, 80, 50);
  }

//----------------SMALLER CLOUD----------------------------X
    for (var i = 0; i < cloud2_x.length; i++) {
      fill(255);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale + increment * 3, 70, 50, 50);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale - 20 + increment * 3, 70, 40, 40);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale - 40 + increment * 3, 70, 20, 20);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale + 25 + increment * 3, 70, 40, 40);

      fill(225);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale + increment * 2, 150, 70, 70);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale - 20 + increment * 2, 150, 60, 60);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale - 40 + increment * 2, 150, 40, 40);
      ellipse(cloud2_x[i].xPos * cloud2_x[i].scale + 25 + increment * 2, 150, 60, 60);
    }
}

function LevelEnd() {
  noStroke();
  fill(13, 42, 117);

  rect(3800, floorPos_y, 500, 150);

  //-----------------LAVA CODE-----------------------------------//

  fill(225, 40, 0);
  rect(3800, floorPos_y + 50 + bop * 0.75, 500 - 10, 100);

  bubblex = random(3800 + 30, 3800 + 500 - 22);
  bubbley = random(floorPos_y + 150, floorPos_y + 70);

  //---------------BUBBLES IN LAVA-----------------------------------------------//

  fill(250, 114, 56);
  circle(bubblex, bubbley, 10);

  //---------------CANYON WALLS-------------------------------------------------//
  fill(80);
  stroke(80);
  rect(3800, floorPos_y, 20, 150);
  rect(3800 + 500 - 10, floorPos_y, 20, 150);

  noStroke();
  fill(127, 127, 127);

  //Castle Wall
  rect(4300, 0, 400, 432);

  stroke(0);

  //Bridge
  fill(188, 156, 102);
  rect(3800, 432, 500, 20);

  //Draw String
  line(4300, 50, 3800, 432);

  //Wheels
  fill(127, 127, 127);
  ellipse(4300, 432, 30, 30);

  fill(188, 156, 102);
  ellipse(4300, 50, 30, 30);
  fill(127, 127, 127);
  ellipse(4300, 50, 10, 10);

  fill(127, 127, 127);
```

```
  fill(127, 127, 127);
  ellipse(3800, 440, 30, 30);
  fill(188, 156, 102);
  ellipse(3800, 440, 10, 10);

  drawBricks();

  //Castle windows
  fill(0);
  arc(-835, 100, 70, 50, PI, TWO_PI);
  rect(-870, 100, 70, 80);
  arc(-565, 100, 70, 50, PI, TWO_PI);
  rect(-600, 100, 70, 80);

  arc(4365, 100, 70, 50, PI, TWO_PI);
  rect(4330, 100, 70, 80);
  arc(4635, 100, 70, 50, PI, TWO_PI);
  rect(4600, 100, 70, 80);
}

//Bricks of the castlewall
function drawBricks() {
  drawbirck();

  for (let i = 0; i < 8; i++) {
    for (let j = 0; j < 10; j++) {
      image(bricks, 4300 + i * 50, floorPos_y - 50 - j * 50, 50, 50);
    }
  }

  for (let i = 0; i < 8; i++) {
    for (let j = 0; j < 15; j++) {
      image(bricks, -900 + i * 50, floorPos_y - 50 - j * 50, 50, 50);
    }
  }
}
function drawTrees() {
  //-------------------------------------TREE-----------------------------//

  //----------FOR LOOP FOR TREE REPEAT------------X

  for (var i = 0; i < trees_x.length; i++) {
    //----------------------TRUNK-------------------------------------------X

    noStroke();
    fill(139, 69, 19); // color of tree trunk

    noStroke();
    rect(trees_x[i].xPos * trees_x[i].scale + 212, 288, 33, 144);

    //---------------TREE__TRIANGLE_BRANCHES--------------------------------X

    fill(0, 80, 0);
    triangle(
      trees_x[i].xPos * trees_x[i].scale + 180,
      370, //left angle
      trees_x[i].xPos * trees_x[i].scale + 280,
      370, // right angle
      trees_x[i].xPos * trees_x[i].scale + 230,
      260
    ); // Apex

    fill(0, 100, 0); //tree branch color
    triangle(
      trees_x[i].xPos * trees_x[i].scale + 180,
      340, //left angle
      trees_x[i].xPos * trees_x[i].scale + 280,
      340, //right angle
      trees_x[i].xPos * trees_x[i].scale + 230,
      230
    ); //Apex

    fill(0, 120, 0); //tree branch color
    triangle(
      trees_x[i].xPos * trees_x[i].scale + 190,
      300, //left angle
      trees_x[i].xPos * trees_x[i].scale + 270,
```

```
      300, //right angle
      trees_x[i].xPos * trees_x[i].scale + 230,
      230
    ); //Apex

    //-------------------------------------------------------------------------X
  }
}



function drawGrass() {
  //-------------------------------------TREE-----------------------------//

  //-----------FOR LOOP FOR GRASS REPEAT-------------X

  for (var i = 0; i < grass2_x.length; i++) {
    image(grass2, grass2_x[i], floorPos_y - 150, 150, 150);
  }

  for (var i = 0; i < grass3_x.length; i++) {
    image(grass3, grass3_x[i], floorPos_y - 130, 130, 130);
  }

  for (var i = 0; i < grass_x.length; i++) {
    image(grass, grass_x[i], floorPos_y - 50, 50, 50);
  }

  dog();
}

//---------------------------------DOG CHARACTER--------------------------//
var doggo_x = 550;

let movingRight = false; // Initialize outside of the function

function dog() {
  let doggo_y = floorPos_y - 100;
  push(); // Save current transformation matrix

  if (doggo_x <= 550 && doggo_x > 0 && !movingRight) {
    doggo_x -= 0.5;
    if (doggo_x == 0) {
      movingRight = true;
    }
    // Draw the dog image normally
    image(doggo, doggo_x, doggo_y);
  } else if (movingRight) {
    doggo_x += 0.5;
    if (doggo_x == 550) {
      movingRight = false;
    }
    translate(doggo_x + doggo.width, doggo_y);
    scale(-1, 1);
    image(doggo, 0, 0);
  }

  pop(); // Restore the transformation matrix
}
//---------------------------MOONs and friend--------------------------------//

function drawMoon() {
  image(moon, gameChar_x + 200, 30, 150, 150);
  image(planet, gameChar_x - 400, 150, 150, 150);
}

function drawFriend() {
  image(friend, gameChar_x - 60, gameChar_y - 100 + cloudbop, 50, 50);
}

function drawMoutains() {
  //-------------------------------MOUTAINS--------------------------------//

  for (var i = 0; i < moutain_x.length; i++) {
    //-------------------------TRIANGLES----------------------------------X
    noStroke();
    fill(57, 67, 92);
    triangle(
```

```
      moutain_x[i].xPos * moutain_x[i].scale + 100,
      258,
      moutain_x[i].xPos * moutain_x[i].scale + 300,
      432,
      moutain_x[i].xPos * moutain_x[i].scale,
      432
    ); // first Shadow of the moutain

    fill(70, 82, 112);

    triangle(
      moutain_x[i].xPos + 100 * moutain_x[i].scale,
      258,
      moutain_x[i].xPos + 200 * moutain_x[i].scale,
      432,
      moutain_x[i].xPos * moutain_x[i].scale,
      432
    ); // first Triangle of the moutain

    fill(70, 82, 112);
    triangle(
      moutain_x[i].xPos + 200 * moutain_x[i].scale,
      198,
      moutain_x[i].xPos + 400 * moutain_x[i].scale,
      432,
      moutain_x[i].xPos + 40 * moutain_x[i].scale,
      432
    ); // Second Triangle of the moutain

    fill(57, 67, 92);
    triangle(
      moutain_x[i].xPos + 200 * moutain_x[i].scale,
      198,
      moutain_x[i].xPos + 450 * moutain_x[i].scale,
      432,
      moutain_x[i].xPos + 330 * moutain_x[i].scale,
      432
    ); // Second shadow of the moutain
  }
}
//-------INTERACTABLES----------------------//
function drawCollectible(collectable) {
  stroke(0);
  fill(218, 165, 32);

  ellipse(
    collectable.x_pos,
    collectable.y_pos + bop,
    35 * collectable.size,
    40 * collectable.size
  );

  fill(255, 215, 0);
  ellipse(
    collectable.x_pos,
    collectable.y_pos + bop,
    25 * collectable.size,
    30 * collectable.size
  );

  fill(218, 165, 32);
  rect(
    collectable.x_pos - 1,
    collectable.y_pos - 7 + bop,
    4 * collectable.size,
    20 * collectable.size
  );
}

function checkCollectable(collectable) {
  if (
    dist(gameChar_x, gameChar_y, collectable.x_pos + 1, collectable.y_pos) < 35
  ) {
    collectable.isFound = true;
    Found.play();
    game_score += 1;
  }
}
```

```
function drawCanyon(canyon) {
  // Draw the canyon and lava here, but remove the falling check.
  noStroke();
  fill(13, 42, 117);

  //----------------BACK OF CANYON---------------------------------//

  rect(canyon.x_pos + 30, floorPos_y, canyon.width - 20, 150);

  //----------------LAVA CODE---------------------------------//

  fill(225, 40, 0);
  rect(canyon.x_pos + 31, floorPos_y + 50 + bop * 0.75, canyon.width - 22, 100);

  bubblex = random(canyon.x_pos + 30, canyon.x_pos + canyon.width - 22);
  bubbley = random(floorPos_y + 150, floorPos_y + 70);

  //----------------BUBBLES IN LAVA---------------------------------//

  fill(250, 114, 56);
  circle(bubblex, bubbley, 10);

  //---------------CANYON WALLS---------------------------------//

  fill(80);
  stroke(80);
  rect(canyon.x_pos + 10, floorPos_y, 20, 150);
  rect(canyon.x_pos + canyon.width - 10, floorPos_y, 20, 150);
}

function drawbirck() {
  noStroke();
  fill(204, 0, 255, 50);

  //Roof
  image(bricks, 460, 210, 85, 20);

  //Energy field
  rect(460, 230, 85, 98);
  fill(216, 130, 237);
  rect(random(460, 545), random(230, 330), 5, 10);

  //Bottom
  image(bricks, 460, 330, 85, 20);

  //-----------------------------------|

  //Bottom
  image(bricks, 1007, 210, 75, 20);
  image(bricks, 1082, 210, 60, 20);

  //Energy Field
  fill(204, 0, 255, 50);
  rect(1007, 230, 135, 98);
  fill(216, 130, 237);
  rect(random(1007, 1130), random(230, 310), 5, 10);

  //Bottom
  image(bricks, 1007, 330, 75, 20);
  image(bricks, 1082, 330, 60, 20);

  //-----------------------------------|

  //roof
  image(bricks, 1757, 210, 75, 20);
  image(bricks, 1832, 210, 75, 20);

  //Energy Field
  fill(204, 0, 255, 50);
  rect(1757, 230, 150, 100);
  fill(216, 130, 237);
  rect(random(1757, 1900), random(230, 310), 5, 10);

  //bottom
  image(bricks, 1757, 330, 75, 20);
  image(bricks, 1832, 330, 75, 20);
}
```

```
function sign() {
    fill(255);
    text("Keep Going ...", 3800, 250);

    if(gameChar_x > 3700) {
        text("I must reach the castle!",gameChar_x + 20, 370);
    }
}

function drawEnemies() {
  for (var i = 0; i < enemies.length; i++) {
    enemies[i].draw();
    var enemiesContact = enemies[i].checkContact(gameChar_x, gameChar_y);

    if (enemiesContact) {
      manabarWidth = 0;
      if (lives < 0) {
        die = true;
        break;
      }
    }
  }
}
function Enemy(x, y, range) {
  this.x = x;
  this.y = y;
  this.range = range;

  this.currentX = x;
  this.inc = 2; //inc for increment
  this.angle = 0;

  //this function increase the value of the enemy by this.inc
  this.update = function () {
    //moves enemies across the screen
    this.currentX += this.inc;

    //if the enemies has gotten to far off right
    if (this.currentX >= this.x + this.range) {
      //move enemy back into position
      this.inc = -2;
    } else if (this.currentX < this.x) {
      //move enemy back into position
      this.inc = 2;
    }
  };

  //draws the enemies-----------------------//
  this.draw = function () {
    //Enemy is a unstable monster that shakes and vibrates
    let shakyEnemy = random(0, 3);
    //draw enemy in updated location
    this.update();

    //The enemy will be angry and shake when player is close
    fill(255, 0, 0);
    stroke(0);
    //spiky shell of enemy
    push();
    translate(this.currentX + shakyEnemy, this.y - 40 + shakyEnemy);
    rotate(this.angle);
    star(0, 0, 20, 40, 20);
    //yes, the enemy is a evil monster eyeball
    pop();
    image(
      eye,
      this.currentX - 39 + shakyEnemy,
      this.y - 91 + shakyEnemy,
      80,
      80
    );
    this.angle += 1;
  };

  //checks the contact with enemy-----------//
  this.checkContact = function (gc_x, gc_y) {
    //checks distance between gamecharx, gamechary and player
```

```
    var EnemyDist = dist(gc_x, gc_y, this.currentX, this.y);

    //Return true if distance between enemy and player below 10
    if (EnemyDist < 40) {
      close = true;
      return true;
    }
    return false;
  };
}

function star(x, y, radius1, radius2, npoints) {
  let angle = TWO_PI / npoints;
  let halfAngle = angle / 2.0;
  beginShape();
  for (let a = 0; a < TWO_PI; a += angle) {
    let sx = x + cos(a) * radius2;
    let sy = y + sin(a) * radius2;
    vertex(sx, sy);
    sx = x + cos(a + halfAngle) * radius1;
    sy = y + sin(a + halfAngle) * radius1;
    vertex(sx, sy);
  }
  endShape(CLOSE);
}

function HUD() {
  //Give background to HUD
  fill(0);
  noStroke();
  rect(gameChar_x - 512, 0, 10000, 60);
  rect(-1000, 0, 10000, 60);

  //-------SCORE HUD--------------------------X
  fill(255);
  stroke(0);
  text("score: " + game_score, cameraPosX + 10, 40);
  //----------------------------------------x

  healthbar();
  manabar();

  //---------RESPAWN TEXT-------//
  if (respawnCooldown == true && !die) {
    //---------Text to display where character has Died----------------//
    fill(255);
    stroke(0);
    textSize(32);
    textFont("comic sans MS");
    text("Respawning...", gameChar_x, 300);

    //-----------------------------------------------------------------//

    //If player presses both left and right together
    if (keyIsDown(LEFT_ARROW) && keyIsDown(RIGHT_ARROW)) {
      text("Both keys are pressed", gameChar_x, 300);
    }
  }
}

//--------CONTROLS MESSAGE------------------//
function control() {
  if (gametime >= 500 && !controlflypast) {
    stroke(0);
    strokeWeight(3);
    textSize(25);
    textFont("comic sans MS");
    fill(255);
    text(
      "Use the arrow keys to control",
      -increment * (-0.08 * increment),
      100
    );
    //so that the player can read the instructions, then zoom off
  }

  if (gametime > 10000) {
    controlflypast = true; //make the control message disappear after 10 seconds
```

```
  }
}

function manabar() {
  if (!isContact && manabarWidth > 0) {
    manabarWidth -= decrement * 0.8;
    red_Increase += 0.5 * 0.7;
    blue_Increase -= 0.5 * 0.7;
  }
  if (isContact && manabarWidth < 200) {
    manabarWidth += decrement * 2;
    red_Increase -= 1;
    blue_Increase += 1;
  }
  if (manabarWidth < 1) {
    //---DISABLE USER INPUT WHEN FALLING------//
    isLeft = false;
    isRight = false;
    isPlummeting = true;
    //---------APPLY GRAVITY TO FALL----------//
    gravity = gravity * 1.1;
    gameChar_y += gravity;
    //---------DEDUCT LIVES-------------------//
    // Only trigger losing a life if not already decremented
    if (!livesDecremented) {
      lose = true;
      livesDecremented = true;
      respawnCooldown = true;
      loseSound.play();
      isFalling = false;
      setTimeout(function () {
        Respawning.play(); //waits for lose sound to complete playing
      }, 1500);

      setTimeout(function () {
        //waits for respawn before setting mana back full
        manabarWidth = 200;
        red_Increase = 0;
        green_Increase = 0;
        blue_Increase = 255;
      }, 5000);
    }
  }
  noStroke();
  fill(255);
  text("Mana", gameChar_x - 180, 40);
  fill(red_Increase, green_Increase, blue_Increase);
  rect(gameChar_x - 100, 20, manabarWidth, 20);
}

//------------------------------HEALTH BAR-------------------------//
function healthbar() {
  fill(0);
  if (lives == 3) {
           image(health, cameraPosX + 800, -10, 200, 80);
  } else if (lives == 2) {
           image(health, cameraPosX + 800, -10, 200, 80);
           noStroke();
           fill(0);
    rect(cameraPosX + 800, 0, 70, 60);
  } else if (lives == 1) {
    image(health, cameraPosX + 800, -10, 200, 80);
           noStroke();
           fill(0);
    rect(cameraPosX + 800, 0, 120, 60);
  } else if (die == true) {
    image(health, cameraPosX + 800, -10, 200, 80);
           noStroke();
           fill(0);
    rect(cameraPosX + 800, 0, 200, 60);
    bg.stop();
  }
}
//--KEYFUCTIONS----------------------------X

function keyPressed() {
  if (keyCode == 37 && lose == false && die == false) {
    isLeft = true;
```

```
  } else if (keyCode == 39 && lose == false && die == false) {
    isRight = true;
  }
  //-------------------------JUMP CODE-----------------------------------X
  else if (
    keyCode == 38 &&
    (isJump == false) & (isFalling == false) &&
    lose == false
  ) {
    isJump = true;
    jumpSound.play();
  } else if (keyPressed == 37 && keyIsDown(RIGHT_ARROW)) {
    stand = true;
    isLeft = true;
    isRight = true;
  }
}
function keyReleased() {
  // if statements to control the animation of the character when
  // keys are released.
  if (keyCode == 37) {
    isLeft = false;
  } else if (keyCode == 39) {
    isRight = false;
  }
}


//--------MECHANICS--------------------------//
function checkCanyon(canyon) {
  if (
    gameChar_x > canyon.x_pos + 55 &&
    gameChar_x < canyon.x_pos + canyon.width - 33 &&
    gameChar_y >= floorPos_y &&
    !livesDecremented
  ) {
    //---DISABLE USER INPUT WHEN FALLING------//
    isLeft = false;
    isRight = false;
    isPlummeting = true;

    //---------APPLY GRAVITY TO FALL----------//
    gravity = gravity * 1.1;
    gameChar_y += gravity;

    //---------DEDUCT LIVES-------------------//

    // Only trigger losing a life if not already decremented
    if (!livesDecremented && !die) {
      lose = true;
      livesDecremented = true;
      respawnCooldown = true;
      loseSound.play();
      setTimeout(function () {
        Respawning.play(); //waits for lose sound to complete playing
      }, 1500);
    }
  }
}

function checkPlayerDie() {
  if (lose && lives > 0 && livesDecremented == true) {
    lives -= 1; //decrement lives by 1

    //-------------RESET BOOLEAN FLAGS----------//

    lose = false;
    isFalling = false;

    //---------Text to display where character has Died-------------------//
    fill(255);
    textSize(32);
    textFont("comic sans MS");
    //------------------------------------------------------------------//
    //---------COOLDOWN TIMER-------------------//

    setTimeout(function () {
      respawn(); // Call respawn function to reset the player state
```

```
      livesDecremented = false; // Reset the flag after respawning
      respawnCooldown = false;
      manabarWidth = 200;
      red_Increase = 0;
      green_Increase = 0;
      blue_Increase = 255;
    }, 5000); //responds in 3 seconds
  } else if (lives <= 0) {
    die = true;

    textSize(30);
    fill(0, 0, 0, 90);
    rect(-2000, 0, 10000, height);
    fill(255);
    text("GAME OVER", cameraPosX + 400, 300);
    const h1Elements = document.getElementsByTagName("h1");

    // Assuming there's only one h1 element
    if (h1Elements.length > 0) {
      h1Elements[0].style.display = "block";
    }
    const reloadElement = document.getElementById("reloadElement");
    reloadElement.addEventListener("click", function() {
      window.location.reload();
    });
  }
}

function respawn() {
  gameChar_x = width / 2;
  gameChar_y = 432;
  cameraPosX = 0; //reset camera position
  gravity = 2; // Reset gravity
  isPlummeting = false;
  isFalling = false;
}

//------------------------------FLAG POLE-------------------------//
function renderFlagpole() {
  push();
  strokeWeight(5);
  stroke(180);
  line(flagpole.x_pos, floorPos_y, flagpole.x_pos, floorPos_y - 250);

  fill(132, 0, 255);
  noStroke();

  if (flagpole.isReached == true) {
    fill(random(0, 255), random(0, 255), 255);
    rect(flagpole.x_pos, floorPos_y - 250, 70, 50);
    image(doggo, flagpole.x_pos - 10, floorPos_y - 305);
  } else {
    fill(0);
    rect(flagpole.x_pos, floorPos_y - 50, 70, 50);
    image(doggo, flagpole.x_pos - 10, floorPos_y - 100);
  }
}

function checkFlagpole() {
  var d = abs(gameChar_x - flagpole.x_pos);

  if (d < 15) flagpole.isReached = true;
}
//-------Win Sdreen--------------//
function playerWin() {
    if(gameChar_x > 4350){
    textSize(30);
    fill(0, 0, 0, 90);
    rect(-2000, 0, 10000, height);
    fill(255);
    text("YOU WIN", cameraPosX + 400, 300);
    noLoop();
    }
}
//------------------------------INTERACTION FUNCTION------------//

function interaction() {
  //--Prevent a bug where character doesnt plummet when in canyon--//
```

```
if (isPlummeting) {
  gravity = gravity * 1.1;
  gameChar_y += gravity;
  return;
}

//--------------------------------WALK-----------------------------------X

if (gameChar_y > 434 && (isFalling == false || isPlummeting == false)) {
  gameChar_y = floorPos_y; //this is to prevent a bug where sometimes the character falls through the floor slowly
}

if (isLeft == true && (isJump == true || isFalling == true)) {
  gameChar_x -= 5;
}

if (isRight == true && (isJump == true || isFalling == true)) {
  gameChar_x += 5;
}

//comments: This is to prevent the walkbop animation from activating while jumping for while falling

//TO PREVENT A BUG WERE IF LEFT AND RIGHT BOTH PRESSED, CAMERAPOSX BREAKS------X
if (isRight == true && isLeft == true) {
  isRight = false;
  isLeft = false;
  cameraPosX = gameChar_x - 512;
}
//-----------------------------------ANIMATED__WALK--------------------X

if (isLeft == true && isJump == false && isFalling == false) {
  gameChar_x -= 5;
  //    gameChar_y = gameChar_y + walkbop; //for walking animation
}

if (isRight == true && isJump == false && isFalling == false) {
  gameChar_x += 5;
  gameChar_y = gameChar_y + walkbop; //for walking animation
}

//---------------------------------MAX HEIGHT-------------------------X

if (isJump == true) {
  gameChar_y -= 10;

  //Max Height Jumping from ground
  if (gameChar_y < 300) {
    isJump = false;
    isFalling = true;
  }
}

//Once player is in the air, it is not in contact
if (gameChar_y <= floorPos_y) {
  isContact = false;

  for (var i = 0; i < platforms.length; i++) {
    //-----------------------------------------------------//
    //If checkContact returns true, player is on platform, after passing X and Y axis check at platform.js line 21
    if (platforms[i].checkContact(gameChar_x_world, gameChar_y) == true) {
      isContact = true; //if player on platform, isContact is true

      isFalling = false; //and is not falling

      break;
    }
  }

  //If not on platform, but on ground
  if (!isContact && gameChar_y < 432) {
    gameChar_y += gravity * 1.3;
    isFalling = false;
  }
}
//-----------------------------------GRAVITY---------------------------X
else if (gameChar_y < floorPos_y) {
  isJump = false;
```

116 lines of code per page

```
    gameChar_y += gravity;
  }

  //-------------------------------------JUMP RIGHT-------------------------X
  else if (gameChar_y < floorPos_y && isRight == true) {
    isRight = true;
    isJump = true;
  }

  //-------------------------------------FALL RIGHT-------------------------X
  else if (isJump && isRight == true) {
    isRight = true;
    isJump = false;
    isFalling = true;
  }

  //-------------------------------------FALL LEFT-------------------------X
  else if (isJump && isLeft == true) {
    isLeft = true;
    isJump = false;
    isFalling = true;
  }

  //-------------------------------------HAS FALLEN GROUND----------------X
  else if (gameChar_y >= floorPos_y) {
    isFalling = false;
  }

  //-------------------------------------JUMP LEFT-------------------------X
  else if (gameChar_y < floorPos_y && isLeft == true) {
    isLeft = true;
    isJump = true;
  }

  //-------------------------------------FLAG POLE-------------------------X
  if (flagpole.isReached == false) {
    checkFlagpole();
  }

  //---------PREVENT DOUBLE JUMPS
  if (gameChar_y < 431 && !Inline) {
    isFalling = true;
  }

  //-------------------------------------Character disappears and do not respawn------------X

  if (die == true) {
    gameChar_y = 1000;
  }
}

function animations() {
  bop = 5 * sin(frameCount / 10); //for the collectible to bop up and down
  walkbop = 2 * sin(frameCount / 3); //for walking animation to bop
  Lavabopbop = 3 * sin(frameCount / 5); //for the lava to rise and fall
  cloudbop = 8 * sin(frameCount / 30); //for the cloud to rise and fall

  //HUD BLACK BAR----------------------------//
  strokeWeight(2);
  fill(77, 114, 84);
  rect(0, floorPos_y, width, height - floorPos_y);
}

let hasAudioPlayed = false;

//     TAKES INPUT FROM PLATFROM ARRAY AT startgame.js line 128
//-----------------------------------------------------------------//
function Platform(x, y, length) {
  this.x = x;
  this.y = y;
  this.length = length;

  this.draw = function () {
    fill(255, 0, 255, 0);
    rect(this.x + 60, this.y, this.length - 120, 20);
  };

  this.drawDebug = function () {
```

```
    noFill();
    stroke(255, 0, 0, 0);
    rect(this.x, this.y, this.length, 20);
  };
  this.checkContact = function (gc_x, gc_y) {


    //X-AXIS CONTACT CHECK------------//

    if (gc_x + 10 > this.x && gc_x < this.x + this.length - 10) {
      Inline = true;

      //Left Threshold check: this.x
      //Right Threshold check: this.x + this.length + 2

      //Y-AXIS CONTACT CHECK------------//
      var d = this.y - gc_y; //distance between p0layer and platform y pos
      if (d >= 0 && d < 5) {
        return true;
      }
    }

    return false;
  };
}
//-----------------------------------------------------------------//

function checkIfCharacterIsOnAnyPlatform() {
  if (isFalling && Inline) {
    for (var i = 0; i < platforms.length; i++) {
      isContact = platforms[i].checkContact(gameChar_x_world, gameChar_y);
      if (isContact == true) {
        onPlatforms = true;
        break;
      }
    }
  }
  // IF character on platform Jumps
  if (!isContact && gameChar_y < 431) {
    isFalling = true;
  }
}

function drawPlatforms() {
  for (var i = 0; i < platforms.length; i++) {
    platforms[i].draw();
    platforms[i].drawDebug(); // This line draws the debug boxes
  }
}

function createPlatform(x, y, length) {
  return new Platform(x, y, length);
}

function preload() {
  jumpSound = loadSound("Sounds/Jump.mp3");
  loseSound = loadSound("Sounds/Lose.mp3");
  Found = loadSound("Sounds/Found.mp3");
  bg = loadSound("Sounds/bg.mp3");
  Birds = loadSound("Sounds/Birds.mp3");
  Death = loadSound("Sounds/Death.mp3");
  Respawning = loadSound("Sounds/Respawning.mp3");
  win = loadSound("Sounds/win.mp3");
  grass = loadImage("Photos/grass.png");
  grass2 = loadImage("Photos/Grass2.png");
  grass3 = loadImage("Photos/Grass3.png");
  doggo = loadImage("Photos/doggo.png");
  moon = loadImage("Photos/moon.png");
  friend = loadImage("Photos/friend.png");
  planet = loadImage("Photos/planet.png");
  bricks = loadImage("Photos/bricks.png");
  portal = loadSound("Sounds/portal.mp3");
  eye = loadImage("Photos/eye.png");
  health = loadImage("Photos/healthbar.png");
}

preload();
```

```
function setup() {
  createCanvas(1024, 576);



  //-------START GAME FUNCTION CALL----------//
  startGame();

  bg.play();
  Birds.play();
  Birds.setVolume(2);


}

function draw() {
  background(13, 42, 117);

  isContact = false;
  Inline = false;
  isFalling = false;

  animations();

  //CAMERA---------------------------------//

  if (isRight == true) {
    cameraPosX += 5;
  } else if (isLeft == true) {
    cameraPosX -= 5;
  }
  push();
  translate(-cameraPosX, 0);

  //------------TRACK PLAYERWORLD POSITION------//
  gameChar_x_world = gameChar_x + cameraPosX;

  //----DRAW ENVIRONMENT----//
  drawStars();
  drawMoon();
  drawClouds();
  drawMoutains();
  drawTrees();
  drawGrass();

  //-----DRAW COIN----------//

  for (var i = 0; i < collectables.length; i++) {
    if (!collectables[i].isFound) {
      checkCollectable(collectables[i]);
      drawCollectible(collectables[i]); // Always draw collectable unless it is found
    }
  }
  //-----DRAW CANYON-------//

  for (var i = 0; i < canyons.length; i++) {
    checkCanyon(canyons[i]);
    drawCanyon(canyons[i]);
  }
  control();
  drawCharacter();
  drawEnemies();
  drawFriend();
  drawBricks();
  LevelEnd();
  renderFlagpole();
  drawPlatforms();
  sign();
  checkIfCharacterIsOnAnyPlatform();
  pop();
  interaction();
  checkPlayerDie();
  HUD();
  playerWin();
}

function startGame() {
  //--------------------POSITIONING VALUES--------------------------------//
```

116 lines of code per page

```
floorPos_y = (height * 3) / 4;
gameChar_x = width / 2;
gameChar_y = floorPos_y - 3;

//---------------------BOOLEAN FLAGS: GAME ENVIRONMENT MECHANICS---------//

gravity = 3;
isLeft = false;
isRight = false;
isFalling = false;
isJump = false;
onPlatforms = false;

//------------------------------MOUTAIN SETUP--------------------------//
moutain_x = [
  { xPos: -400, scale: 1.5 },
  { xPos: 0, scale: 0.8 },
  { xPos: 200, scale: 1 },
  { xPos: 900, scale: 1 },
  { xPos: 7600, scale: 1 },
  { xPos: 3000, scale: 1 },
  { xPos: 3300, scale: 1 },
];
//------------------------TREE  SETUP-----------------------------------//

trees_x = [
  { xPos: -600, scale: 1 },
  { xPos: -500, scale: 1 },
  { xPos: -400, scale: 1 },
  { xPos: -300, scale: 1 },
  { xPos: -100, scale: 1 },
  { xPos: 50, scale: 1 },
  { xPos: 200, scale: 1 },
  { xPos: 400, scale: 1 },
  { xPos: 700, scale: 1 },
  { xPos: 900, scale: 1 },
  { xPos: 1500, scale: 1 },
  { xPos: 1700, scale: 1 },
  { xPos: 2300, scale: 1 },
  { xPos: 2500, scale: 1 },
  { xPos: 2800, scale: 1 },
  { xPos: 3000, scale: 1 },
  { xPos: 3500, scale: 1 },
];

//------------------------GRASS  SETUP-----------------------------------//
//The flowers on the level
grass_x = [
  -600, -550, -530, -500, -450, -430, -300, -290, -70, -60, 0, 10, 50, 80,
  120, 150, 200, 400, 430, 450, 680, 900, 1000, 1100, 1200, 1300, 1700, 1800,
  1900, 2000, 2450, 2500, 2600, 2700, 3200, 3250, 3300, 3350, 3400, 3700,
];

//Television on the ground because why noy
grass2_x = [400, 1700, 3300];

//Pipe flower
grass3_x = [40, 900, 1300];

//Small castle at end of level
castleX = [3500];

//---------------------CAYON  SETUP-----------------------------------//

canyons = [
  { x_pos: -1700, width: 700 },
  { x_pos: 700,   width: 200 },
  { x_pos: 1400,  width: 300 },
  { x_pos: 2100,  width: 250 },
  { x_pos: 2800,  width: 150 },
];

//------------------------COLLECTABLE  SETUP-----------------------------//

//default for size is 0.7;

collectables = [
  { x_pos: 300,  y_pos: 405, size: 0.7, isFound: false },
```

```
    { x_pos: 540,  y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 800,  y_pos: 300, size: 0.8, isFound: false },
    { x_pos: 1000, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 1200, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 1400, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 1550, y_pos: 300, size: 0.8, isFound: false },
    { x_pos: 1700, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 1900, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 2100, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 2400, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 2600, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 2800, y_pos: 405, size: 0.7, isFound: false },
    { x_pos: 2900, y_pos: 300, size: 0.8, isFound: false },
    { x_pos: 3100, y_pos: 405, size: 0.7, isFound: false },
  ];

  //-----------------------CLOUD  SETUP---------------------------------//

  cloud_x = [
    { xPos: 200,   scale: 1 },
    { xPos: 600,   scale: 1 },
    { xPos: 900,   scale: 1 },
    { xPos: 1300,  scale: 1 },
    { xPos: 1600,  scale: 1 },
    { xPos: 2000,  scale: 1 },
    { xPos: 2500,  scale: 1 },
    { xPos: 3000,  scale: 1 },
    { xPos: 3500,  scale: 1 },
    { xPos: 4000,  scale: 1 },
    { xPos: 4500,  scale: 1 },
    { xPos: 5000,  scale: 1 },
    { xPos: 5500,  scale: 1 },
    { xPos: 6000,  scale: 1 },
    { xPos: 6500,  scale: 1 },
    { xPos: 7000,  scale: 1 },
    { xPos: 7500,  scale: 1 },
    { xPos: 8000,  scale: 1 },
    { xPos: 8500,  scale: 1 },
    { xPos: 9000,  scale: 1 },
    { xPos: 9500,  scale: 1 },
    { xPos: 10000, scale: 1 },
    { xPos: 10500, scale: 1 },
    { xPos: 11000, scale: 1 },
    { xPos: 11500, scale: 1 },
    { xPos: 12000, scale: 1 },
    { xPos: 12500, scale: 1 },
    { xPos: 13000, scale: 1 },
    { xPos: 13500, scale: 1 },
    { xPos: 14500, scale: 1 },
    { xPos: 15000, scale: 1 },
    { xPos: 15500, scale: 1 },
    { xPos: 16000, scale: 1 },
    { xPos: 16500, scale: 1 },
    { xPos: 17000, scale: 1 },
    { xPos: 17500, scale: 1 },
    { xPos: 18000, scale: 1 },
    { xPos: 18500, scale: 1 },
    { xPos: 19000, scale: 1 },
    { xPos: 19500, scale: 1 },
  ];

  //For the smaller clouds
cloud2_x = [
  { xPos: 200, scale: 1 }, { xPos: 600, scale: 1 }, { xPos: 900, scale: 1 },
  { xPos: 1300, scale: 1 }, { xPos: 1600, scale: 1 }, { xPos: 2000, scale: 1 },
  { xPos: 2500, scale: 1 }, { xPos: 3000, scale: 1 }, { xPos: 3500, scale: 1 },
  { xPos: 4000, scale: 1 }, { xPos: 4500, scale: 1 }, { xPos: 5000, scale: 1 },
  { xPos: 5500, scale: 1 }, { xPos: 6000, scale: 1 }, { xPos: 6500, scale: 1 },
  { xPos: 7000, scale: 1 }, { xPos: 7500, scale: 1 }, { xPos: 8000, scale: 1 },
  { xPos: 8500, scale: 1 }, { xPos: 9000, scale: 1 }, { xPos: 9500, scale: 1 },
  { xPos: 10000, scale: 1 }, { xPos: 105000, scale: 1 }, { xPos: 11000, scale: 1 },
  { xPos: 11500, scale: 1 }, { xPos: 12000, scale: 1 }, { xPos: 12500, scale: 1 },
  { xPos: 13000, scale: 1 }, { xPos: 13500, scale: 1 }, { xPos: 14500, scale: 1 },
  { xPos: 15000, scale: 1 }, { xPos: 15500, scale: 1 }, { xPos: 16000, scale: 1 },
  { xPos: 16500, scale: 1 }, { xPos: 17000, scale: 1 }, { xPos: 17500, scale: 1 },
  { xPos: 18000, scale: 1 }, { xPos: 18500, scale: 1 }, { xPos: 19000, scale: 1 },
  { xPos: 19500, scale: 1 }
];
```

```
  stars = [];

  //Coin Object-------------//

  coin = {
    coin_Posy: -5,
  };

  //Game Score--------------//

  game_score = 0;

  //Flagpole objects--------//

  flagpole = { isReached: false, x_pos: 3500 };

  //Lives-------------------//

  lives = 3;

  //Platforms---------------//

  //-------------------PLATFORM---------------------------------------//

  platforms = [];
  platforms.push(createPlatform(400, floorPos_y - 100, 200));
  platforms.push(createPlatform(1500, floorPos_y - 100, 200));
  platforms.push(createPlatform(1600, floorPos_y - 100, 200));
  platforms.push(createPlatform(3000, floorPos_y - 100, 200));
  platforms.push(createPlatform(3200, floorPos_y - 100, 200));

  //-------------------PLATFORM---------------------------------------//

  enemies = [];
  enemies.push(new Enemy(100, floorPos_y - 10, 300));
  enemies.push(new Enemy(1000, floorPos_y - 10, 400));
  enemies.push(new Enemy(1600, floorPos_y - 10, 400));
  enemies.push(new Enemy(1400, floorPos_y - 100, 1000));
}

//---------- GLOBAL VARS ----------//

//---- POS VARIABLES ----//
var gameChar_x;
var gameChar_y;
var floorPos_y;
var collectable = 0.7;
var cameraPosX   = 0;

//---- ARRAY POS VARS ----//
var trees_x;
var moutain_x;
var caynon_x;
var Canyon;
var cloud_x;
var grass_x;
var grass3_x;
var grass2_x;
var stars;
var castleX;
var cloud2_x;
var bricks;
var castleWallX;

//---- MOVEMENT VARS ----//
var isLeft;
var isRight;
var isFalling = false;
var isPlummeting;
var isJump;
var stand;

//---- ENV MECHANICS ----//
var platforms;
var Inline;
var onPlatforms;
```

116 lines of code per page

```
var isContact;
var gravity;
var bop;
var walkbop;
var Lavabop;
var cloudbop;
var increment = 1;
var decrement = 0.3;

//---- GAME MECHANICS ----//
var gametime;
var game_score = 0;
var flagpole;
var lives = 3;
var respawnCooldown = 0;

//---- BOOL FLAGS -------//
var ground          = true;
var lose            = false;
var isFound         = false;
var livesDecremented= false;
var die             = false;
var plumetcanyon    = false;
var respawnCooldown = false;
var controlflypast  = false;

//---- ART ELEMENTS ----//
var manabarWidth = 200;
var red_Increase = 0;
var blue_Increase = 255;
var green_Increase = 0;
var bubblex;
var bubbley;

//----- PRELOAD --------//
var moon;
var castle;
var doggo;
var grass2;
var grass3;
//-------ENEMIES--------//
var enemies;
//-------DOM ELEMENTS--//
var newParagraph;
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Path to Freedom</title>
    <link rel="icon" href="../Resources/ManStand.png" type="image/x-icon" />

    <script>
        document.addEventListener("DOMContentLoaded", function() {
            document.getElementById("playButton").addEventListener("click", function() {
                var video = document.getElementById("myVideo");
                video.style.display = "block";
                video.play();
            });
        });
    </script>


    <style>
      body {
        background: black;
      }

      h1 {
        background: green;
        text-align: center;
        padding: 1%;
        color: white;
        font-family: comic sans ms;
      }

      a {
        text-decoration: none;
```

116 lines of code per page

```
      }

      #playButton {
        font-family: comic sans ms;
        font-size: 150%;
        background: blue;
        color: white;
        width: 50%;
        height: 60px;
      }
    </style>
  </head>
  <body>
    <!-- This is the button that will trigger the video to appear and play -->
    <button id="playButton">Play final Cutscene</button>

    <!-- Initially set the video to be hidden using inline style or CSS -->
    <video
      id="myVideo"
      width="90%"
      height="auto"
      controls
      style="display: none"
    >
      <source src="../Resource2/Ends.mp4" type="video/mp4" />
      Your browser does not support the video tag.
    </video>

    <div>
      <a href="index.html">
        <h1>click here for the bonus level</h1>
      </a>
    </div>
  </body>
</html>

<!DOCTYPE html>
<html>
  <head> </head>
  <body>
    <script src="./Libraries/p5.min.js"></script>
    <script src="./Libraries/p5.sound.min.js"></script>
    <script src="preload.js"></script>
    <script src="Draw.js"></script>
    <script src="mechanics.js"></script>
    <script src="sketch.js"></script>
    <script src="Misc.js"></script>
    <script src="key.js"></script>
    <script src="Character.js"></script>
    <script src="startgame.js"></script>
    <script src="platform.js"></script>
    <script src="variables.js"></script>
    <script src="HUD.js"></script>
    <script src="enemies.js"></script>
    <link rel="stylesheet" href="CSS/index.css" />

    <h1 id="reloadElement">Reload</h1>

    <div class="info">
    <p>Use the Arrow Keys to navigate and spacebar to jump</p>
    <p>Stand on the platfrom to recharge your Mana, if your mana falls below 0, character will die</p>
    <p>Avoid the spiky Monsters</p>
    <p>This game may require a little patience, enjoy!</p>
    </div>

    <button onclick="redirectToPage()">Go back to main menu</button>

    <script>
        function redirectToPage() {
            window.location.href = "../Menu.html";
        }
    </script>



  </body>
</html>
```