

Process & Decision Documentation

Project/Assignment Decisions

Modified the code so instead of a win or lose game, it is a beach adventure game where users can choose their own ending.

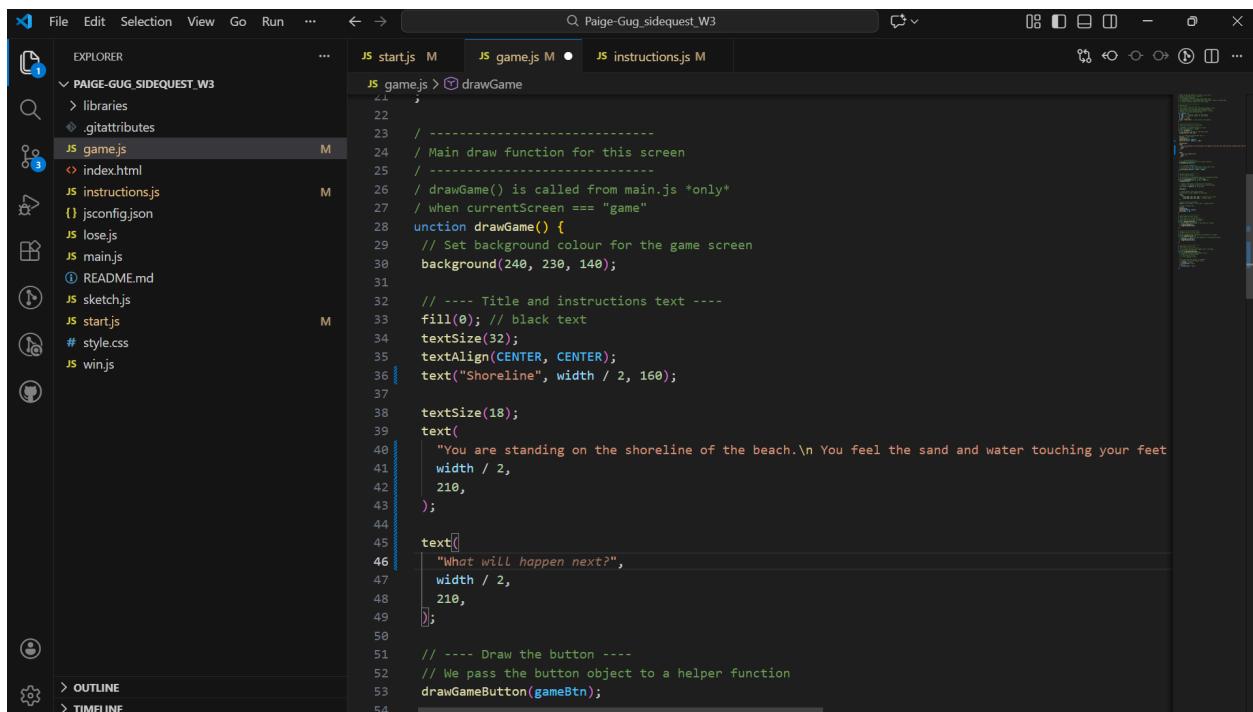
Role-Based Process Evidence

Goal of Work Session

Create a choose your own adventure storyline.

Tools, Resources, or Inputs Used:

- Week 3 example 1 code from GBDA 302 Learn



The screenshot shows a code editor interface with a dark theme. The left sidebar is labeled 'EXPLORER' and lists files in a folder named 'PAIGE-GUG_SIDEQUEST_W3'. The files listed are: start.js (M), game.js (M), instructions.js (M), libraries, .gitattributes, index.html, jsconfig.json, lose.js, main.js, README.md, sketch.js, startjs, # style.css, and winjs. The main editor area has tabs for 'start.js M', 'game.js M' (which is currently active), and 'instructions.js M'. The code in 'game.js' is as follows:

```
JS game.js M JS game.js M JS instructions.js M
JS game.js > drawGame
21;
22;
23; / -----
24; / Main draw function for this screen
25; -----
26; / drawGame() is called from main.js *only*
27; / when currentScreen === "game"
28; unction drawGame() {
29;   // Set background colour for the game screen
30;   background(240, 230, 140);
31;
32;   // ---- Title and instructions text ----
33;   fill(0); // black text
34;   textSize(32);
35;   textAlign(CENTER, CENTER);
36;   text("Shoreline", width / 2, 160);
37;
38;   textSize(18);
39;   text(
40;     "You are standing on the shoreline of the beach.\n You feel the sand and water touching your feet
41;     width / 2,
42;     210,
43;   );
44;
45;   text([
46;     "What will happen next?",
47;     width / 2,
48;     210,
49;   ]);
50;
51;   // ---- Draw the button ----
52;   // We pass the button object to a helper function
53;   drawGameButton(gameBtn);
54;
```

The image shows two screenshots of a code editor interface, likely VS Code, displaying JavaScript files for a game project named "PAIGE-GUG SIDEQUEST_W3".

Screenshot 1: The active tab is `JS game.js`. The code handles button drawing and keyboard input. It includes a function `drawGameButton` and a `gameMousePressed` event handler. The `gameMousePressed` handler checks if the left or right button is pressed and updates the `currentScreen` variable accordingly.

```

    JS game.js > ⚡ gameMousePressed
    72     function drawGameButton({ x, y, w, h, label }) {
    73         : color(200, 220, 255, 190), // normal state
    74     );
    75
    76     // Draw the button rectangle
    77     rect(x, y, w, h, 14); // last value = rounded corners
    78
    79     // Draw the button text
    80     fill(0);
    81     textSize(28);
    82     textAlign(CENTER, CENTER);
    83     text(label, x, y);
    84
    85     // -----
    86     // Keyboard input for this screen
    87     // -----
    88     // Allows keyboard-only interaction (accessibility + design)
    89     function gameMousePressed() {
    90         // ENTER key triggers the same behaviour as clicking the button
    91         if (isHover(leftBtn)) {
    92             currentScreen = "lose"; You, 14 seconds ago * Uncommitted changes
    93         }
    94         if (isHover(rightBtn)) {
    95             currentScreen = "win";
    96         }
    97     }
    98
    99     // -----
    100    // Keyboard input for this screen
    101    // -----
    102    // Allows keyboard-only interaction (accessibility + design)
    103    function gameMousePressed() {
    104        // ENTER key triggers the same behaviour as clicking the button
    105        if (isHover(leftBtn)) {
    106            currentScreen = "lose"; You, 14 seconds ago * Uncommitted changes
    107        }
    108        if (isHover(rightBtn)) {
    109            currentScreen = "win";
    110        }
    111    }
    112
  
```

Screenshot 2: The active tab is `JS start.js`. The code defines a `drawStart` function which sets up the start screen. It draws a background rectangle, title text ("Welcome to the beach!"), and two buttons: "start" and "instructions". The `start` button is located at `x: width / 2, y: 420, w: 240, h: 80`, and the `instructions` button is located at `x: width / 2, y: 530, w: 240, h: 80`.

```

    JS start.js > ⚡ drawStart
    7
    8     // -----
    9     // Start screen visuals
    10    // -----
    11    // drawStart() is called from main.js only when:
    12    // currentScreen === "start"
    13    function drawStart() {
    14        // Background colour for the start screen
    15        background(180, 225, 220); // soft teal background
    16        rect(0, height - height / 4, width, height / 2); // bottom rectangle You, now * Uncommitted changes
    17
    18        // ---- Title text ----
    19        fill(30, 50, 60);
    20        textSize(46);
    21        textAlign(CENTER, CENTER);
    22        text("Welcome to the beach!", width / 2, 280);
    23
    24        // ---- Buttons (data only) ----
    25        // These objects store the position/size/label for each button.
    26        // Using objects makes it easy to pass them into drawButton()
    27        // and also reuse the same information for hover checks.
    28        const startBtn = {
    29            x: width / 2,
    30            y: 420,
    31            w: 240,
    32            h: 80,
    33            label: "Click to explore",
    34        };
    35
    36        const instrBtn = {
    37            x: width / 2,
    38            y: 530,
    39            w: 240,
    40            h: 80
  
```

GenAI Documentation

No GenAI was used for this assignment.