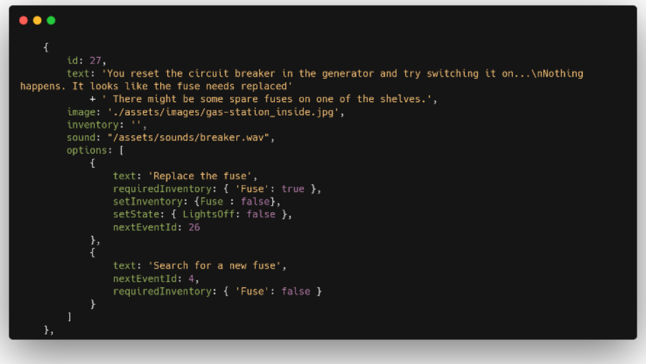
Individual Summary Report

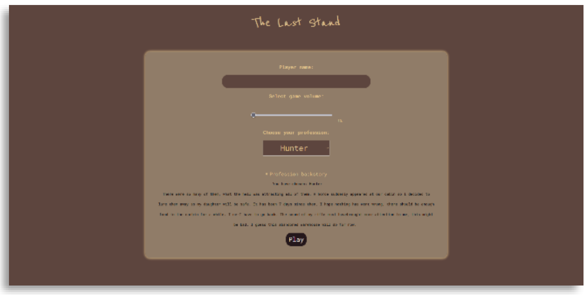
Ryan Beattie

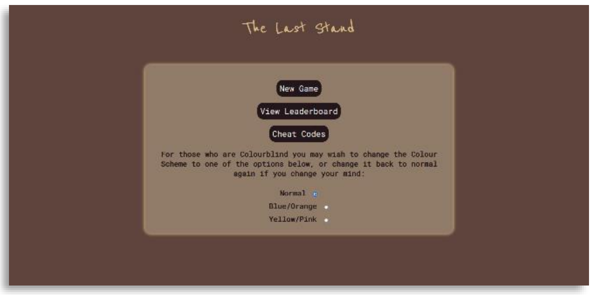
One of the first contributions I made to our project was the text node system to navigate between events/locations in the game. Each text node consists of a unique identifier and several options of which can contain a set of required inventory items or game states. Game state and inventory items can also be set. Each option is displayed as a button and only options where inventory/state requirements are met are displayed to the player. This text node system allowed me to create a basic structure for my group to use for each location and has proven very successful in the context of our game.



```
    {
        id: 27,
        text: 'You reset the circuit breaker in the generator and try switching it on...\nNothing
happens. It looks like the fuse needs replaced'
            + ' There might be some spare fuses on one of the shelves.',
        image: './assets/images/gas-station_inside.jpg',
        inventory: '',
        sound: "/assets/sounds/breaker.wav",
        options: [
            {
                text: 'Replace the fuse',
                requiredInventory: { 'Fuse': true },
                setInventory: {Fuse : false},
                setState: { LightsOff: false },
                nextEventId: 26
            },
            {
                text: 'Search for a new fuse',
                nextEventId: 4,
                requiredInventory: { 'Fuse': false }
            }
        ]
    },
```

One of the other contributions I made in the beginning of the project was the player menu, where the player can enter their name, set the volume of their game and select their profession. These values are passed between locations using sessionStorage. Each profession can have unique events/endings depending on the location. I completed all the HTML and CSS elements of this page, aside from the radio buttons which have recently been added by Andrew for his feature to change the colour scheme. I also completed the JS for this page to switch from the menu screen to the character creation page. This page started with just the option to select the profession but has since developed to include a player name input, which I implemented JS to display in the top left corner of the game screen, and a game volume slider, which adjusts and displays the game volume, and I implemented with help from James. The player's name is required for leader board purposes.

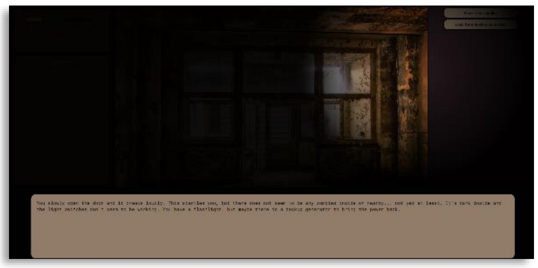




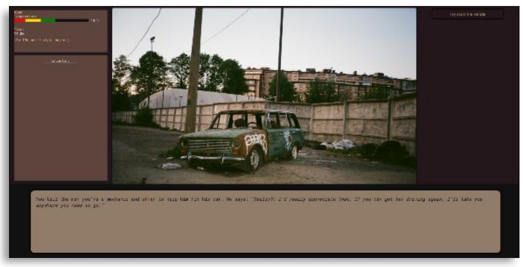
For those who are Colourblind you may wish to change the Colour Scheme to one of the options below, or change it back to normal again if you change your mind:

[✓ Normal]
Blue/Orange
Yellow/Pink

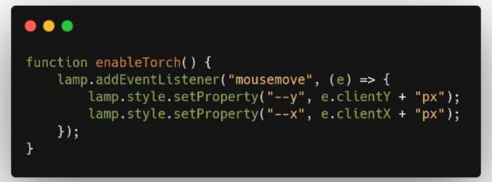
My chosen location for the game is the gas station. Similarly, to the other locations, this location was developed using the text node system. Some unique features I implemented into the gas station location include a CSS torch effect when inside. This can be disabled by searching for a fuse and replacing the broken fuse in the backup generator which can be found underneath the counter. This effect mainly consists of CSS, but some JS is also required to add an event listener and to set when to enable/disable the feature. Additionally, another unique feature I implemented is the ability to win the game without having to complete the preparation phase as seen in other locations. If the chosen profession is mechanic, the player can offer to fix the stranger's car, which later enables them to either leave with the stranger in the car or steal it and leave the stranger for dead at the gas station. Although a relatively small feature, this adds a unique way of winning the game.



You slowly open the door and it creaks loudly. This startles you, but there does not seem to be any zombies inside or nearby... not yet at least. It's dark inside and the light switches don't seem to be working. You have a flashlight, but maybe there is a backup generator to bring the power back.

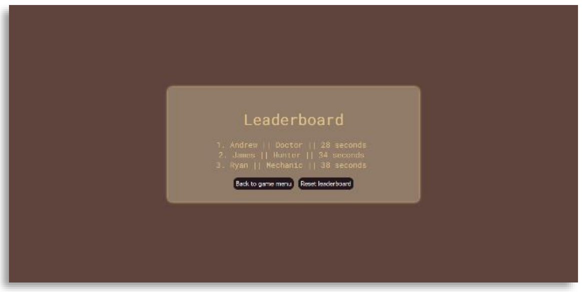


You tell the man you're a mechanic and offer to help him fix his car. He says: "Really?! I'd really appreciate that. If you can get her driving again, I'll take you anywhere you need to go."



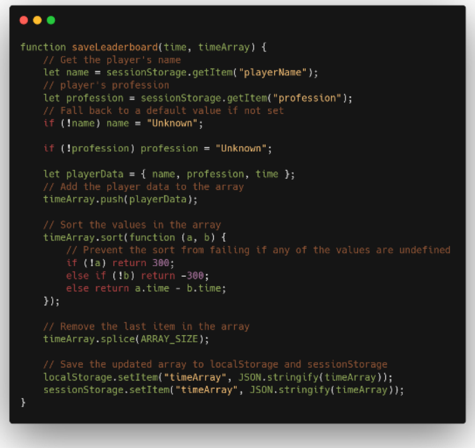
```
function enableTorch() {
    lamp.addEventListener("mousemove", (e) => {
        lamp.style.setProperty("--y", e.clientY + "px");
        lamp.style.setProperty("--x", e.clientX + "px");
    });
}
```

I decided to implement a leader board into the game towards the end of the development of our game. I started by implementing the HTML and CSS of the page. James helped me with the JS to display the name and play time of the shortest playthrough, part of which is shown in the snippet below. I finished the JS by adding a condition so that only winning games are added to the leader board and adding the profession of the player to be displayed along with the name and time. The leader board can be reset by clicking the 'Clear Leader board' button, which I initially set to clear localStorage, however, the JS for clearing the leader board has since been changed by James.



Leaderboard

1. Andrew || Doctor || 28 seconds
2. James || Hunter || 34 seconds
3. Ryan || Mechanic || 38 seconds



```
function saveLeaderboard(time, timeArray) {
    // Get the player's name
    let name = sessionStorage.getItem("playerName");
    // player's profession
    let profession = sessionStorage.getItem("profession");
    // Fall back to a default value if not set
    if (!name) name = "Unknown";

    if (!profession) profession = "Unknown";

    let playerData = { name, profession, time };
    // Add the player data to the array
    timeArray.push(playerData);

    // Sort the values in the array
    timeArray.sort(function (a, b) {
        // Prevent the sort from failing if any of the values are undefined
        if (!a) return 300;
        else if (!b) return -300;
        else return a.time - b.time;
    });

    // Remove the last item in the array
    timeArray.splice(ARRAY_SIZE);

    // Save the updated array to localStorage and sessionStorage
    localStorage.setItem("timeArray", JSON.stringify(timeArray));
    sessionStorage.setItem("timeArray", JSON.stringify(timeArray));
}
```

On an additional note – I created a prototype version for an inventory system for the game, however, the implementation of this feature was later taken over by James.