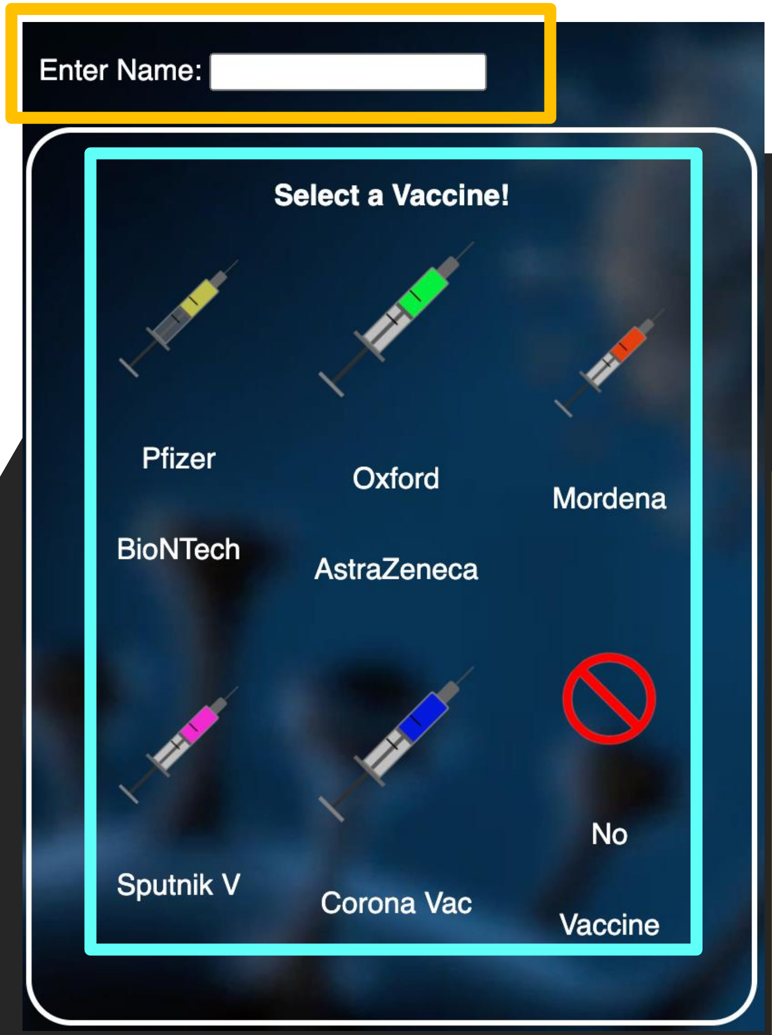
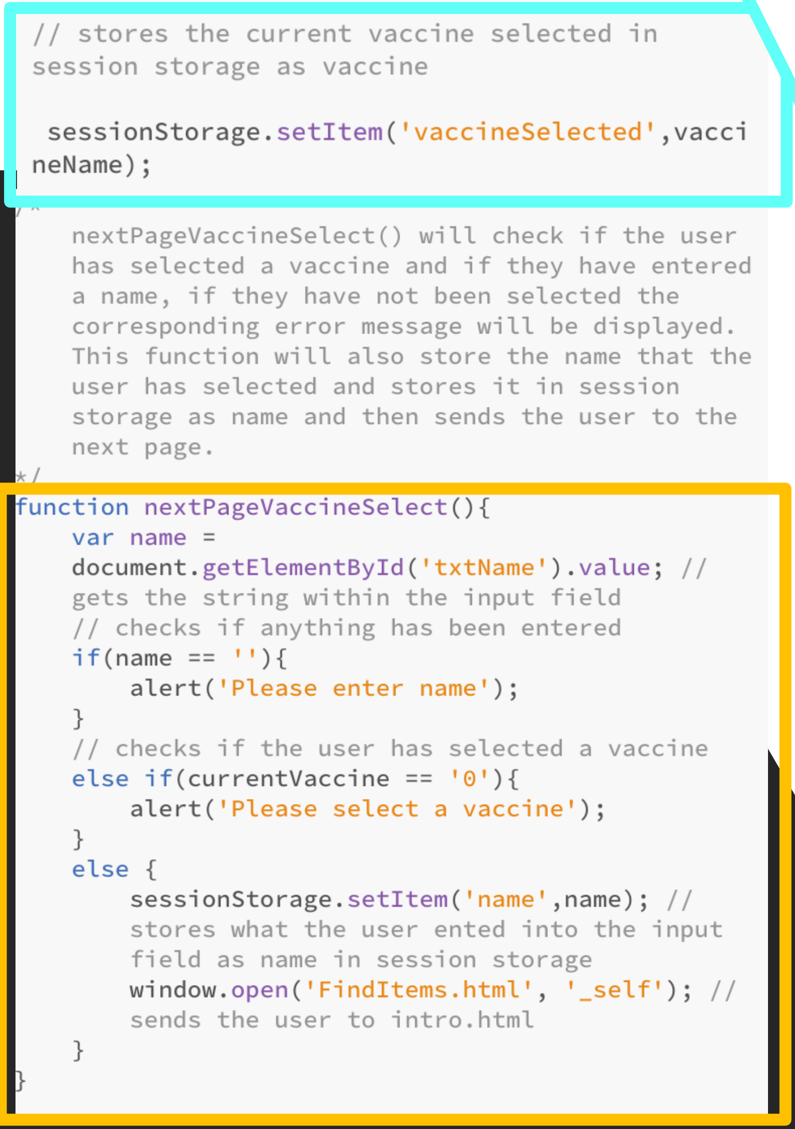
Designed & Implemented by

Adam Logan	40293585
XXXXX	
XXXXX	
XXXXX	
XXXXX	

Data Collection





```
// stores the current vaccine selected in session storage as vaccine

 sessionStorage.setItem('vaccineSelected',vacci
neName);
/*
    nextPageVaccineSelect() will check if the user
    has selected a vaccine and if they have entered
    a name, if they have not been selected the
    corresponding error message will be displayed.
    This function will also store the name that the
    user has selected and stores it in session
    storage as name and then sends the user to the
    next page.
*/
function nextPageVaccineSelect(){
    var name =
    document.getElementById('txtName').value; //
    gets the string within the input field
    // checks if anything has been entered
    if(name == ''){
        alert('Please enter name');
    }
    // checks if the user has selected a vaccine
    else if(currentVaccine == '0'){
        alert('Please select a vaccine');
    }
    else {
        sessionStorage.setItem('name',name); //
        stores what the user ented into the input
        field as name in session storage
        window.open('FindItems.html', '_self'); //
        sends the user to intro.html
    }
}
```

User selects vaccine for each game, and this will be stored in session storage

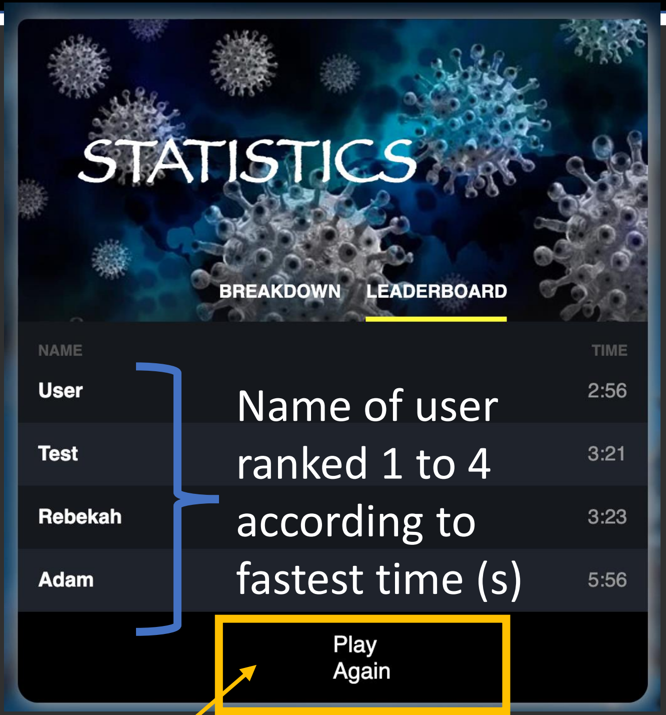
Collect users name at start of COV!D CR!S!S escape room and store it in session storage to be used for personalization of story throughout the game and to store the name associated with the time in the leaderboard

Data Collection

COV!D CR!S!S

Leaderboard

Breakdown

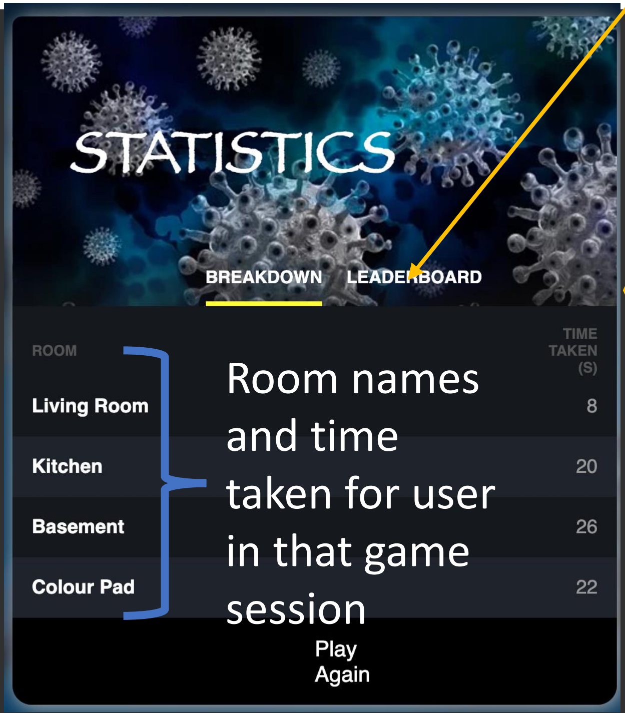


NAME	TIME
User	2:56
Test	3:21
Rebekah	3:23
Adam	5:56

Play Again

Name of user ranked 1 to 4 according to fastest time (s)

User can press this button and escape room again



ROOM	TIME TAKEN (S)
Living Room	8
Kitchen	20
Basement	26
Colour Pad	22

Play Again

Room names and time taken for user in that game session

User can navigate from breakdown and leaderboard using the tags at the top

The Leaderboard collects overall time taken data from each player, stores as local storage and ranks them 1 to 4

The Breakdown collects data on the time taken (seconds) in each room by that individual player as these times have been stored in session storage throughout the game

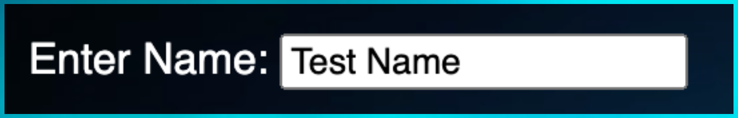
Game Personalization

NAME PERSONALIZATION:

To ensure the users gains a personalized experience whilst playing COV!D CR!S!S we ask the user to enter their name and use it when displaying:

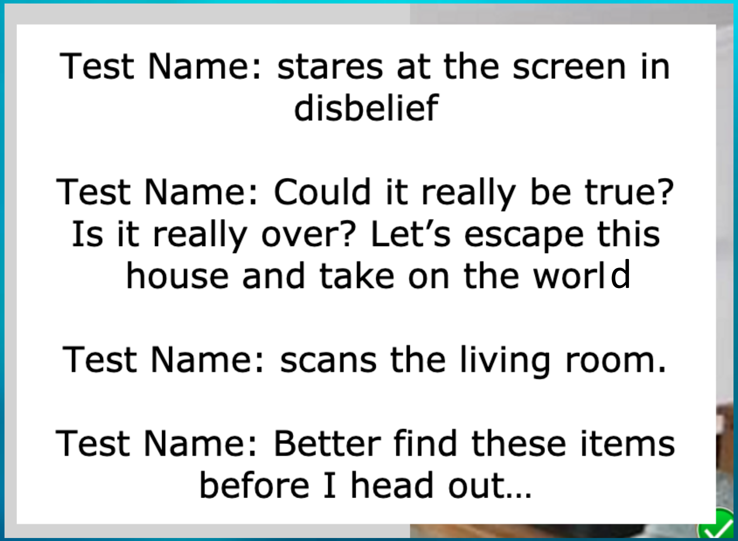
1. Story
2. Instructions
3. Leaderboard Statistics

User asked to enter their name



Enter Name: Test Name

The users name that was entered on the start screen of COV!D CR!S!S is then displayed in all pages that have a story or any instructions displayed. Here is an example from the find items page



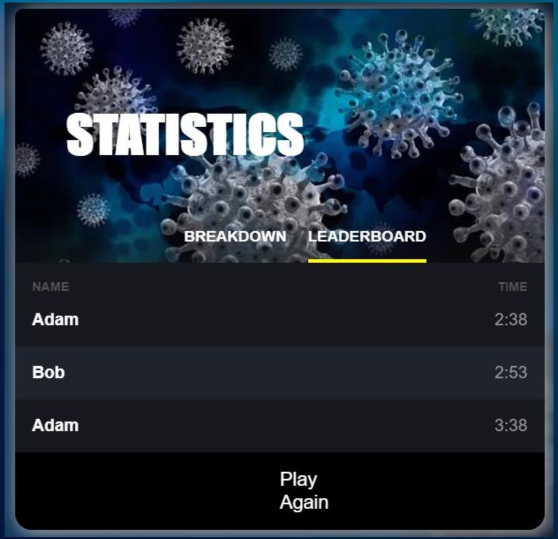
Test Name: stares at the screen in disbelief

Test Name: Could it really be true? Is it really over? Let's escape this house and take on the world

Test Name: scans the living room.

Test Name: Better find these items before I head out…

Once the user completes the escape room the leaderboard is displayed, the users name is displayed in the leaderboard if they have reached the top 4 rank otherwise their name will not be displayed in the leaderboard



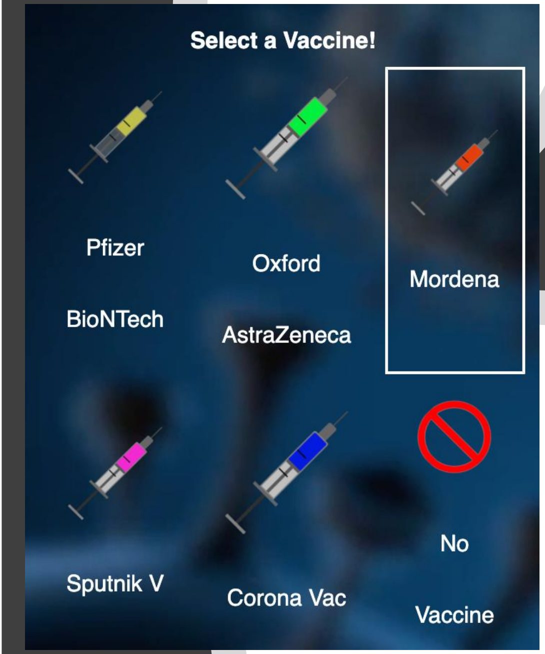
NAME	TIME
Adam	2:38
Bob	2:53
Adam	3:38

Game Personalization

Vaccine personalization:

Once the user is asked for their name they are given the option to choose a vaccine (Pfizer, Oxford, Moderna, Spuntik V, Corona Vac) or choose no vaccine. The vaccine the user chooses will change some options in the game such as the vaccine the user chooses will determine what pointless items they must find. Furthermore, if the user selects no vaccine at the end of the game the user will be tole that they have tested positive for Covid even if they successfully complete the game

User Selects Vaccine


Select a Vaccine!

Pfizer Selected

Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitisier
Visor
Book

Oxford Selected

Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitisier
Visor
Tea Cup

Moderna Selected

Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitisier
Visor
Window

Sputnik V Selected

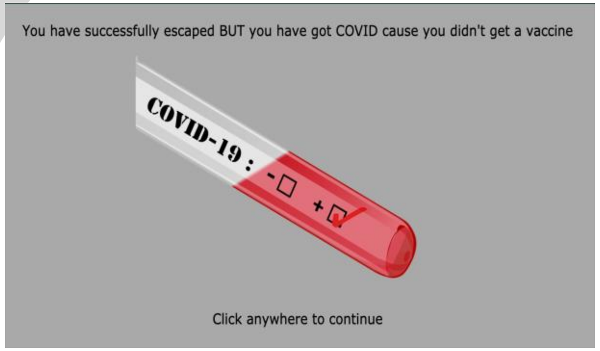
Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitisier
Visor
Hat and Scarf

Corona Vac selected

Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitisier
Visor
Fireplace

No Vaccine selected

Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitisier
Visor


You have successfully escaped BUT you have got COVID cause you didn't get a vaccine

Click anywhere to continue

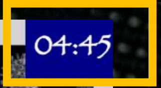
JavaScript Timed Events

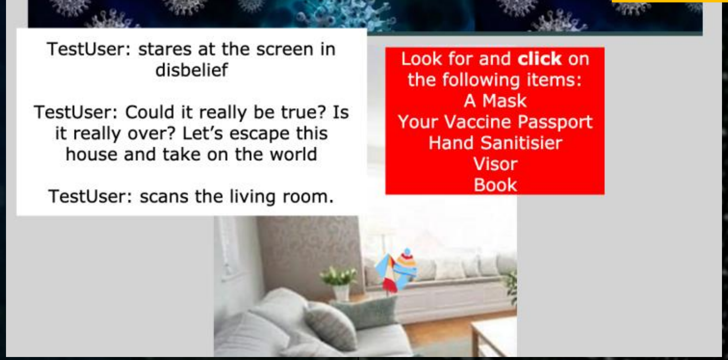
Total 5 Minute Timer

Individual Room Timer

Changing Text According to Timer

Game Timer (5 Mins)



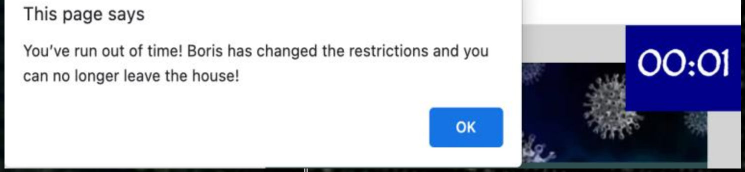

TestUser: stares at the screen in disbelief

TestUser: Could it really be true? Is it really over? Let's escape this house and take on the world

TestUser: scans the living room.

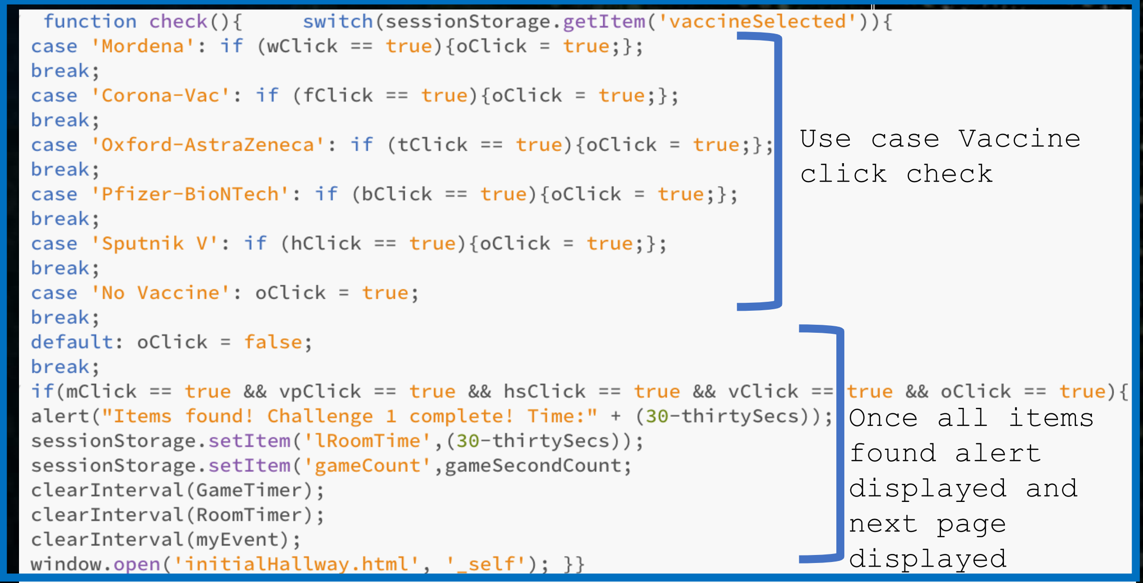
Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitiser
Visor
Book

Example of Timer Running Out


This page says

You've run out of time! Boris has changed the restrictions and you can no longer leave the house!

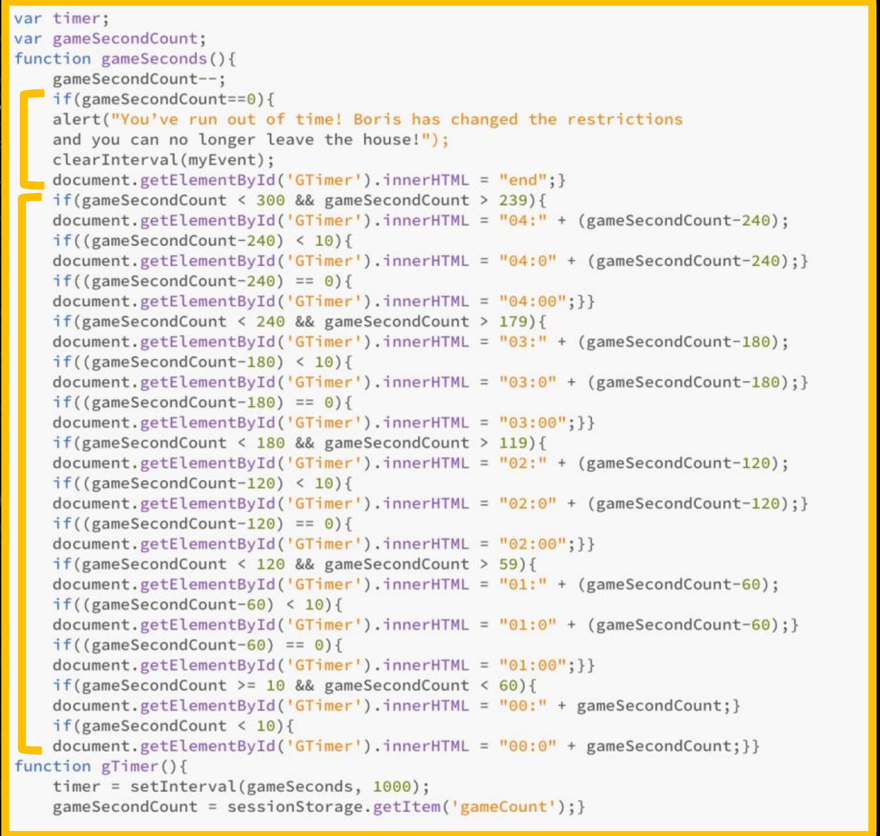
Individual Room Timer Code


```
 function check(){      switch(sessionStorage.getItem('vaccineSelected')){
case 'Mordena': if (wClick == true){oClick = true;};
break;
case 'Corona-Vac': if (fClick == true){oClick = true;};
break;
case 'Oxford-AstraZeneca': if (tClick == true){oClick = true;};
break;
case 'Pfizer-BioNTech': if (bClick == true){oClick = true;};
break;
case 'Sputnik V': if (hClick == true){oClick = true;};
break;
case 'No Vaccine': oClick = true;
break;
default: oClick = false;
break;
if(mClick == true && vpClick == true && hsClick == true && vClick == true && oClick == true){
alert("Items found! Challenge 1 complete! Time:" + (30-thirtySecs));
sessionStorage.setItem('lRoomTime',(30-thirtySecs));
sessionStorage.setItem('gameCount',gameSecondCount;
clearInterval(GameTimer);
clearInterval(RoomTimer);
clearInterval(myEvent);
window.open('initialHallway.html', '_self'); }}
```

Use case Vaccine click check

Once all items found alert displayed and next page displayed

Game Timer Code

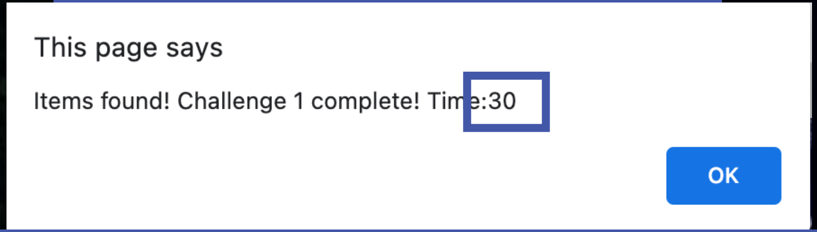

```
var timer;
var gameSecondCount;
function gameSeconds(){
    gameSecondCount--;
    if(gameSecondCount==0){
    alert("You've run out of time! Boris has changed the restrictions
    and you can no longer leave the house!");
    clearInterval(myEvent);
    document.getElementById('GTimer').innerHTML = "end";}
    if(gameSecondCount < 300 && gameSecondCount > 239){
    document.getElementById('GTimer').innerHTML = "04:" + (gameSecondCount-240);
    if((gameSecondCount-240) < 10){
    document.getElementById('GTimer').innerHTML = "04:0" + (gameSecondCount-240);}
    if((gameSecondCount-240) == 0){
    document.getElementById('GTimer').innerHTML = "04:00";}}
    if(gameSecondCount < 240 && gameSecondCount > 179){
    document.getElementById('GTimer').innerHTML = "03:" + (gameSecondCount-180);
    if((gameSecondCount-180) < 10){
    document.getElementById('GTimer').innerHTML = "03:0" + (gameSecondCount-180);}
    if((gameSecondCount-180) == 0){
    document.getElementById('GTimer').innerHTML = "03:00";}}
    if(gameSecondCount < 180 && gameSecondCount > 119){
    document.getElementById('GTimer').innerHTML = "02:" + (gameSecondCount-120);
    if((gameSecondCount-120) < 10){
    document.getElementById('GTimer').innerHTML = "02:0" + (gameSecondCount-120);}
    if((gameSecondCount-120) == 0){
    document.getElementById('GTimer').innerHTML = "02:00";}}
    if(gameSecondCount < 120 && gameSecondCount > 59){
    document.getElementById('GTimer').innerHTML = "01:" + (gameSecondCount-60);
    if((gameSecondCount-60) < 10){
    document.getElementById('GTimer').innerHTML = "01:0" + (gameSecondCount-60);}
    if((gameSecondCount-60) == 0){
    document.getElementById('GTimer').innerHTML = "01:00";}}
    if(gameSecondCount >= 10 && gameSecondCount < 60){
    document.getElementById('GTimer').innerHTML = "00:" + gameSecondCount;}
    if(gameSecondCount < 10){
    document.getElementById('GTimer').innerHTML = "00:0" + gameSecondCount;}}
function gTimer(){
    timer = setInterval(gameSeconds, 1000);
    gameSecondCount = sessionStorage.getItem('gameCount');}
```

If no time left

Display of time

Place in session storage to pass to next page

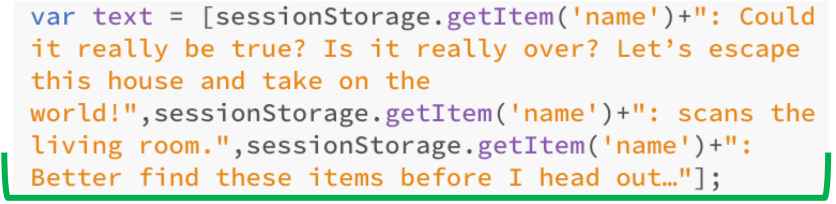
Individual Room Timer


This page says

Items found! Challenge 1 complete! Time:30

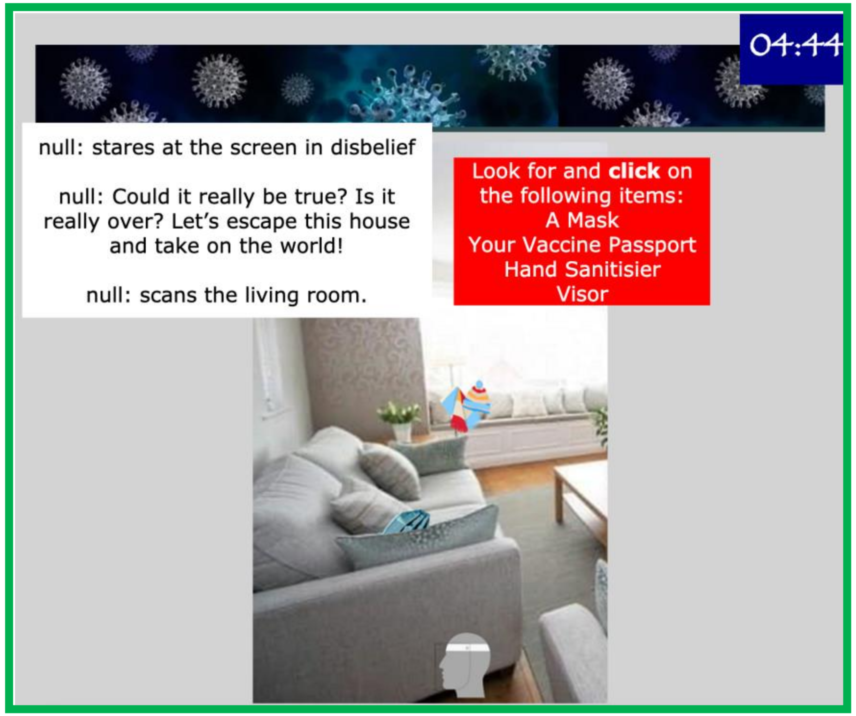
This example shows the time it took to complete the find items challenge

JavaScript Timed Events



```
var text = [sessionStorage.getItem('name')+": Could
it really be true? Is it really over? Let’s escape
this house and take on the
world!",sessionStorage.getItem('name')+": scans the
living room.",sessionStorage.getItem('name')+":
Better find these items before I head out…"];
```

Array of the text that we want to be displayed including the name that was stored in session storage



04:44

null: stares at the screen in disbelief

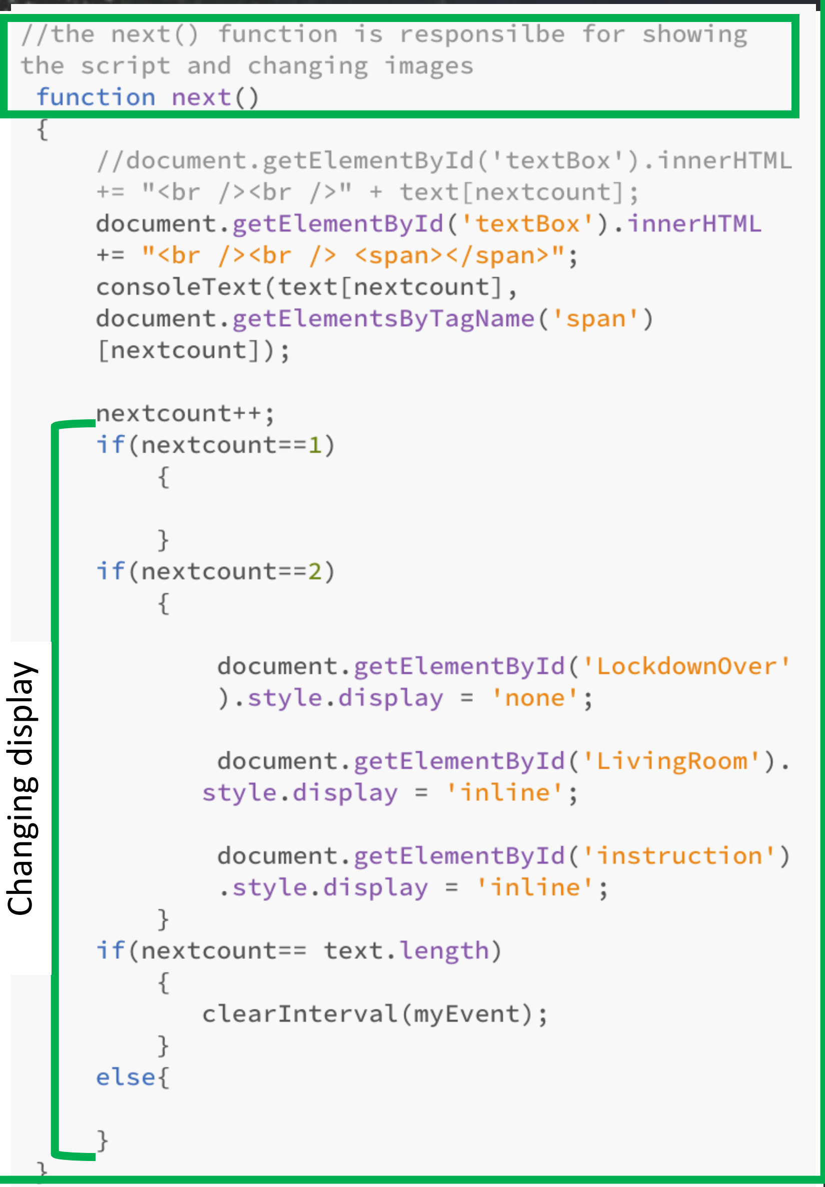
null: Could it really be true? Is it really over? Let’s escape this house and take on the world!

null: scans the living room.

Look for and **click** on the following items:
A Mask
Your Vaccine Passport
Hand Sanitisier
Visor

Here we have inserted a snippet of code from the find items page to show how we change the text alongside the timer. This is achieved by placing our text into an array and then displaying it when the timer reaches a certain amount, we have used this method in multiple ways throughout our entire escape room.

For example, we also use this code to change the image/ page the user is at according the the timer



```
//the next() function is responsilbe for showing
the script and changing images
 function next()
 {
     //document.getElementById('textBox').innerHTML
     += "<br /><br />" + text[nextcount];
     document.getElementById('textBox').innerHTML
     += "<br /><br /> <span></span>";
     consoleText(text[nextcount],
     document.getElementsByTagName('span')
     [nextcount]);

     nextcount++;
     if(nextcount==1)
         {

         }
     if(nextcount==2)
         {

             document.getElementById('LockdownOver'
             ).style.display = 'none';

             document.getElementById('LivingRoom').
            style.display = 'inline';

             document.getElementById('instruction')
             .style.display = 'inline';
         }
     if(nextcount== text.length)
         {
            clearInterval(myEvent);
         }
     else{

     }
 }
```

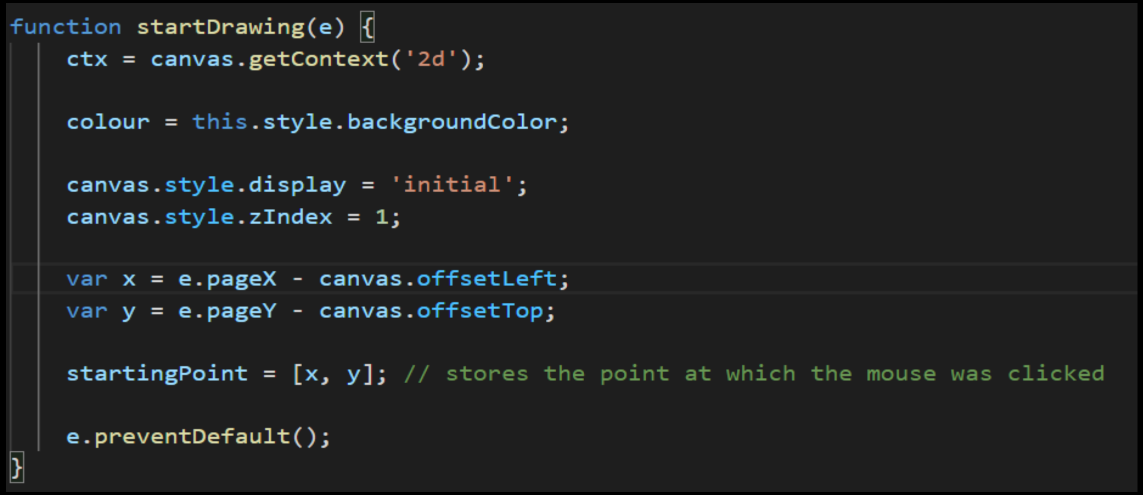
Changing display

Escape Room Elements

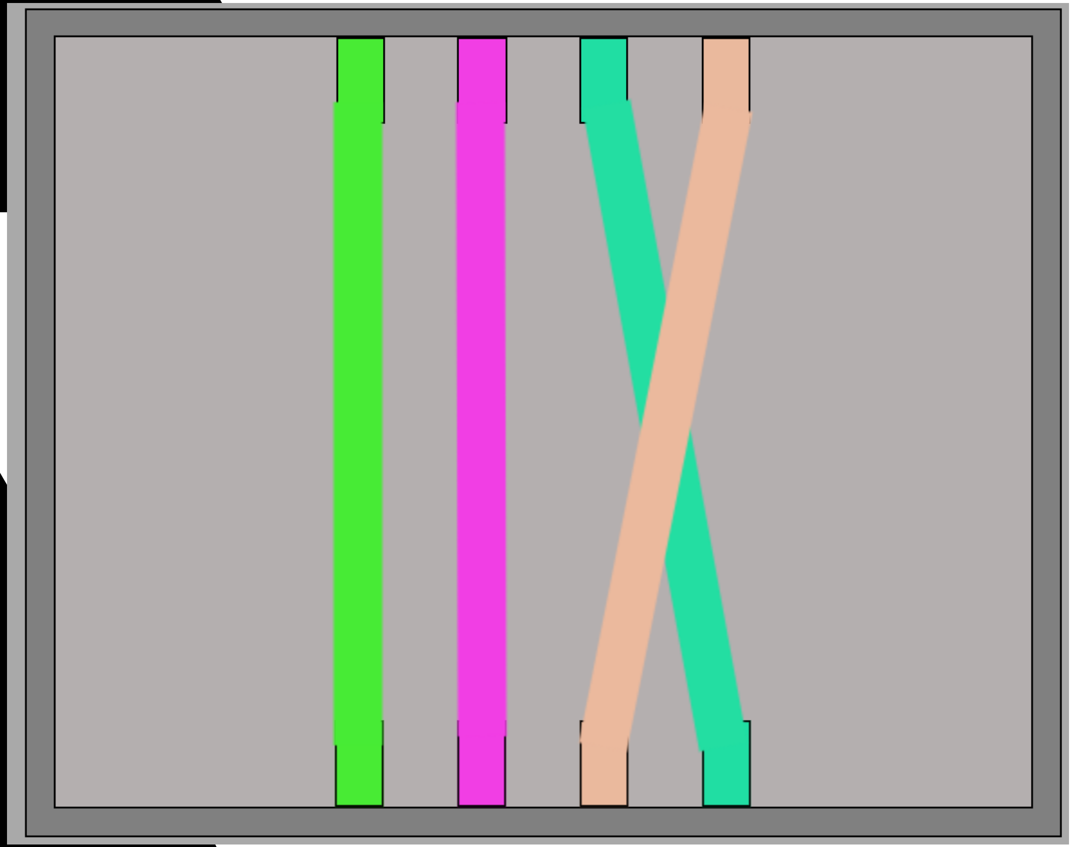
Fuse Box

The fuse box game was created by using the HTML element ‘canvas’ to draw on the web page. The colours of the wires are randomly generated and are checked if they are similar to the other wire colours by checking how close the RGB values are. The position of the start and end connectors are also random, so the corresponding connectors are different for each playthrough. All the code to create the fuse box game could not fit onto this slide so the function to begin the drawing is shown below but if you wish to see the code in its entirety just go to fuseboxScript.js.

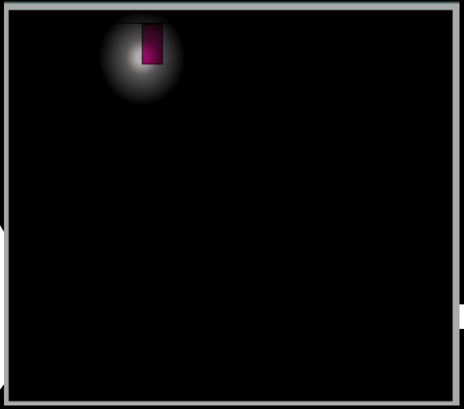
The function which starts the drawing of the wire:



```
function startDrawing(e) {
    ctx = canvas.getContext('2d');

    colour = this.style.backgroundColor;

    canvas.style.display = 'initial';
    canvas.style.zIndex = 1;

    var x = e.pageX - canvas.offsetLeft;
    var y = e.pageY - canvas.offsetTop;

    startingPoint = [x, y]; // stores the point at which the mouse was clicked

    e.preventDefault();
}
```

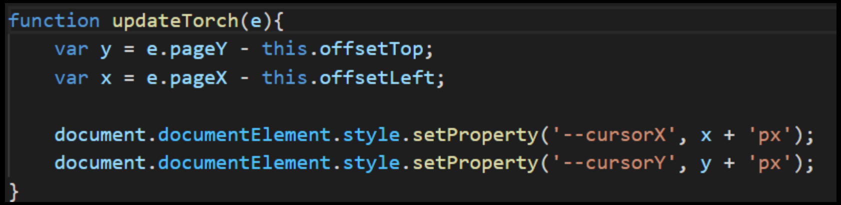


Torch Effect



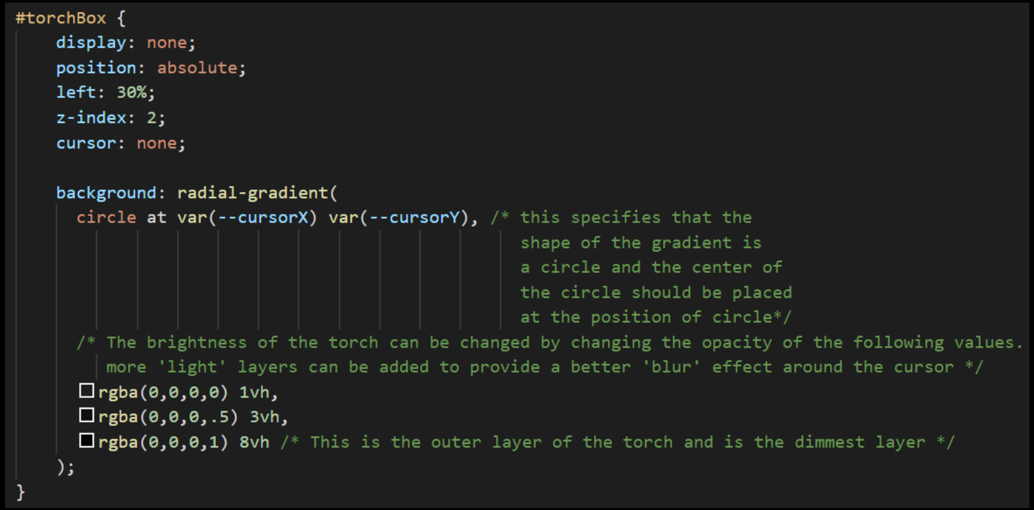
The torch effect has been implemented using custom CSS properties to store the position of the cursor in a way in which a CSS file can access them. The radial-gradient function within CSS has been used in conjunction within the background property to create this effect.

JavaScript Code:



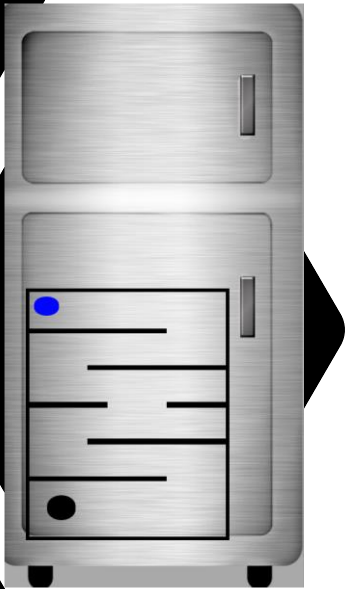
```
function updateTorch(e){
    var y = e.pageY - this.offsetTop;
    var x = e.pageX - this.offsetLeft;

    document.documentElement.style.setProperty('--cursorX', x + 'px');
    document.documentElement.style.setProperty('--cursorY', y + 'px');
}
```

CSS Code:



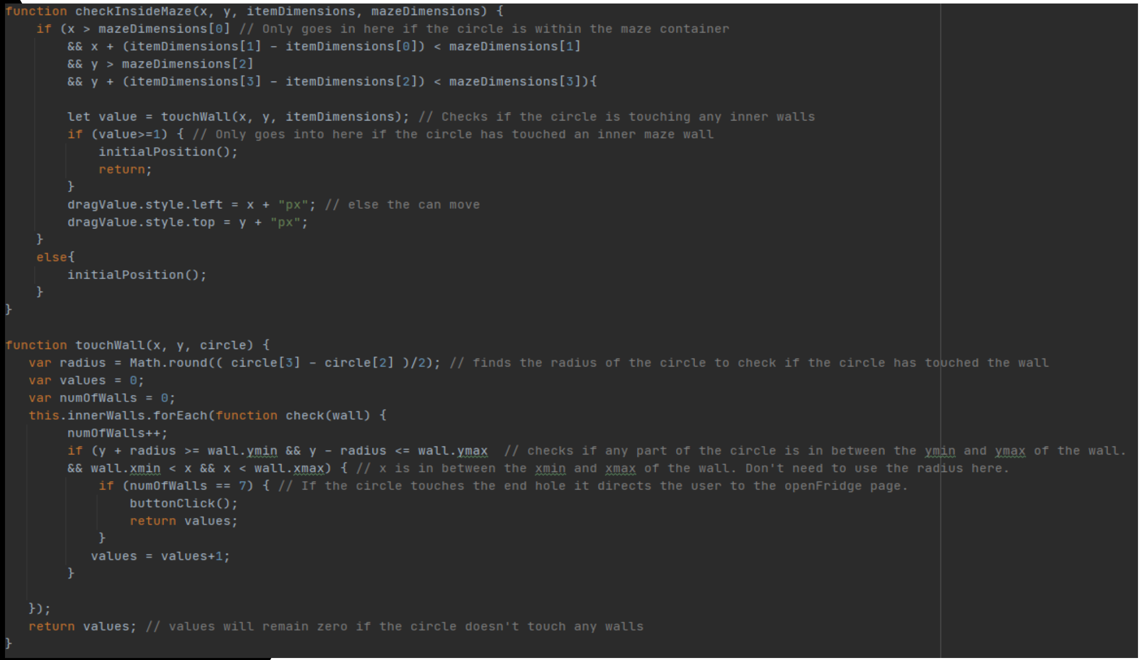
```
#torchBox {
    display: none;
    position: absolute;
    left: 30%;
    z-index: 2;
    cursor: none;

    background: radial-gradient(
      circle at var(--cursorX) var(--cursorY), /* this specifies that the
                                                 shape of the gradient is
                                                 a circle and the center of
                                                 the circle should be placed
                                                 at the position of circle*/
      /* The brightness of the torch can be changed by changing the opacity of the following values.
         more 'light' layers can be added to provide a better 'blur' effect around the cursor */
      □rgba(0,0,0,0) 1vh,
      □rgba(0,0,0,.5) 3vh,
      □rgba(0,0,0,1) 8vh /* This is the outer layer of the torch and is the dimmest layer */
    );
}
```

Fridge Maze

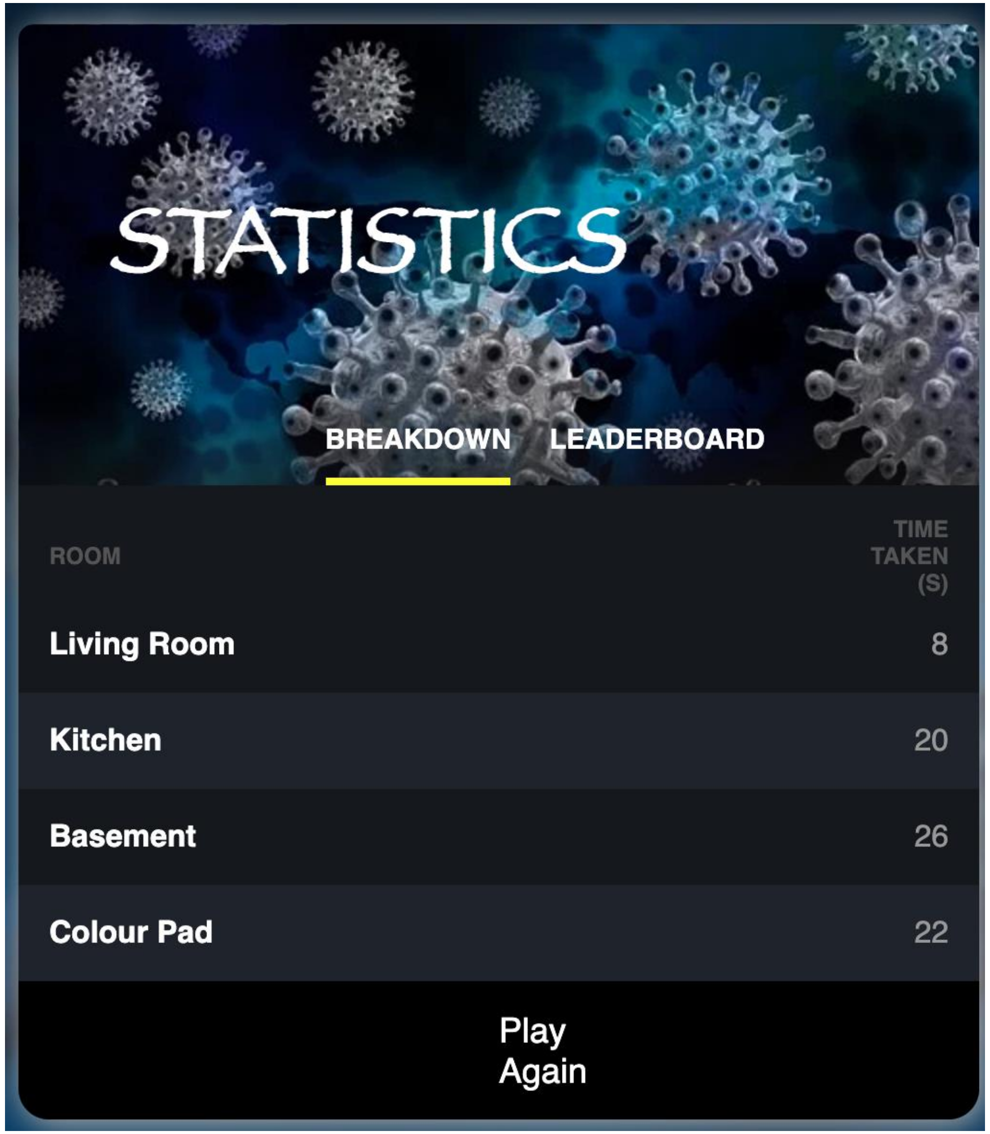


The maze game has been implemented using a draggable item with restrictive outer maze borders and restrictive inner maze walls. The draggable item returns to the initial position if it touches any restrictive boundary.

JavaScript Code:



```
function checkInsideMaze(x, y, itemDimensions, mazeDimensions) {
    if (x > mazeDimensions[0] // Only goes in here if the circle is within the maze container
        && x + (itemDimensions[1] - itemDimensions[0]) < mazeDimensions[1]
        && y > mazeDimensions[2]
        && y + (itemDimensions[3] - itemDimensions[2]) < mazeDimensions[3]){

        let value = touchWall(x, y, itemDimensions); // Checks if the circle is touching any inner walls
        if (value>=1) { // Only goes into here if the circle has touched an inner maze wall
            initialPosition();
            return;
        }
        dragValue.style.left = x + "px"; // else the can move
        dragValue.style.top = y + "px";
    }
    else{
        initialPosition();
    }
}

function touchWall(x, y, circle) {
   var radius = Math.round(( circle[3] - circle[2] )/2); // finds the radius of the circle to check if the circle has touched the wall
   var values = 0;
   var numOfWalls = 0;
   this.innerWalls.forEach(function check(wall) {
        numOfWalls++;
        if (y + radius >= wall.ymin && y - radius <= wall.ymax  // checks if any part of the circle is in between the ymin and ymax of the wall.
        && wall.xmin < x && x < wall.xmax) { // x is in between the xmin and xmax of the wall. Don't need to use the radius here.
            if (numOfWalls == 7) { // If the circle touches the end hole it directs the user to the openFridge page.
                buttonClick();
                return values;
            }
          values = values+1;
        }

   });
   return values; // values will remain zero if the circle doesn't touch any walls
}
```



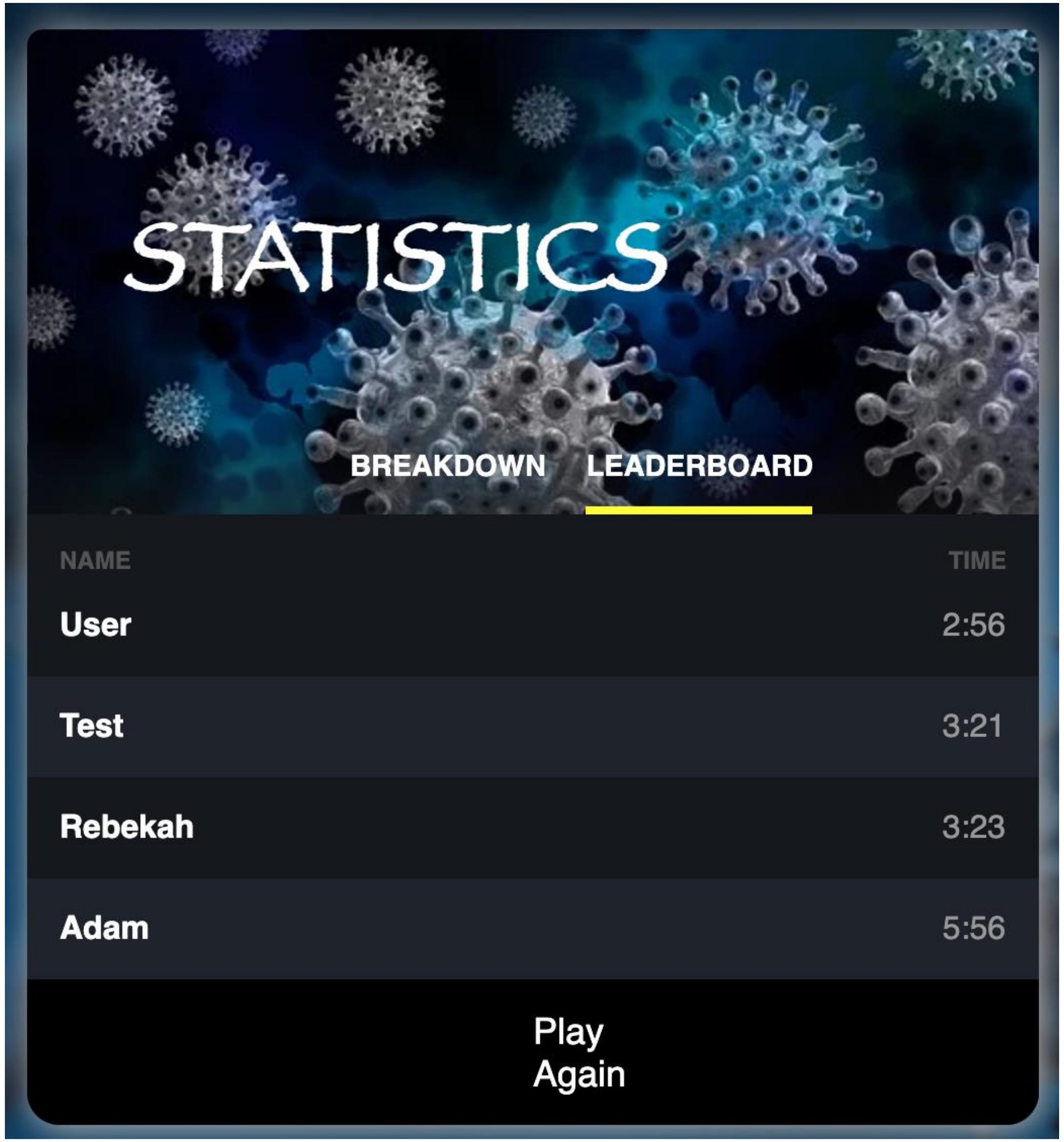
ROOM	TIME TAKEN (S)
Living Room	8
Kitchen	20
Basement	26
Colour Pad	22

Statistics Generated

Here we have inserted a screenshot of our breakdown Statistics. This is the page that the user is shown once they finish the escape room. The breakdown displays all the stored time taken from each room individually

- living room (find items challenge)
- kitchen (fridge maze challenge)
- basement (torch, fuse box and coat code)
- colour pad (hex code challenge)

The breakdown statistics were derived by firstly reading the time taken in each room (once challenge room opens until challenge room closes) and then storing it as session storage so it can be passed and called again, then we displayed this time in the time taken on the table.



NAME	TIME
User	2:56
Test	3:21
Rebekah	3:23
Adam	5:56

Statistics Generated

COV!D CR!S!S

The leaderboard displays a rank of the top 4 players that completed the escape room the quickest along side their name and time taken.

If the user did not place in the top 4 then their time taken total will not be shown on the leaderboard

We feel like the leaderboard is very important as it makes the game more competitive

The leaderboard statistics are derived by using a timer. The 5-minute timer begins to count down as soon as the escape room starts, the timer is displayed in the top right corner of all pages. The timer then counts down from 5 minutes, if the user does not finish the game after 5 minute the user has lost. However, if the user does finish the game in under 5 minutes the leaderboard will be displayed. The overall time left is passed through the pages using local storage. We then calculate the time taken out of the 5 minutes using the time left and display it in the table.

Accessibility Features

COV!D CR!S!S

CONTRAST –ALLOWING USER TO CHANGE CONTRAST MEANS THAT THEY CAN CHOOSE A COLOUR SCHEME THAT CAUSES LESS STRAIN ON THEIR EYES

FONT SIZE- INCREASING THE FONT SIZE HELPS TO AVOID STRAINING YOUR EYES FROM SQUINTING AND FOR THOSE WITH COMPROMISED VISION, THIS CAN MAKE READING MUCH EASIER

COLOUR BLIND PAD - CREATING A COLOUR BLIND BUTTON ON THE COLOUR PAD MEANS THAT THE USER WILL NOT BE AT A DISADVANTAGE AS THE NAME OF THE COLOUR LINKED TO THE HEX CODE WILL BE DISPLAYED INSTEAD OF THE COLOUR

The user can accessibility options from the start screen :

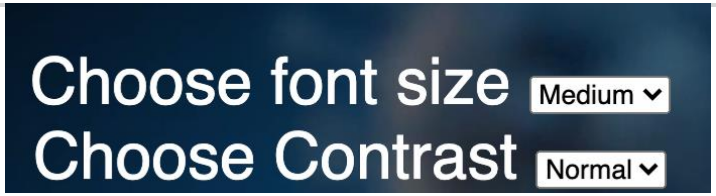
1. Font size

- ○ Small
- ○ Medium
- ○ Large

2. Contrast

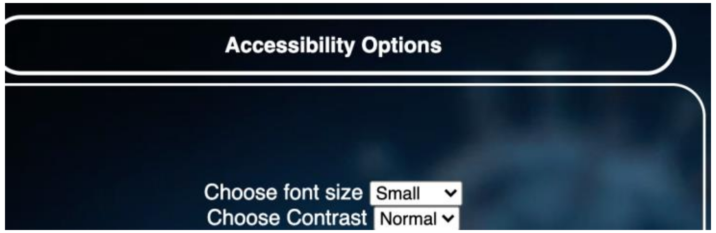
- ○ Normal
- ○ High

We have also added vibrant colours, clear text, clear images and audio throughout our game to ensure that it is not strenuous on the and stimulating with sounds such as wires buzzing or typing when messages display, this ensures that the users has a good experience



Choose font size Medium
Choose Contrast Normal

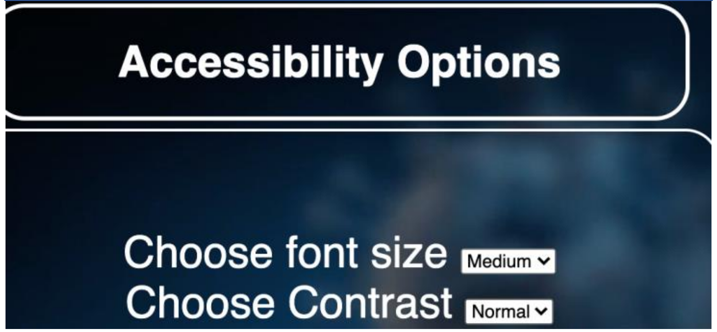
Small Font



Accessibility Options

Choose font size Small
Choose Contrast Normal

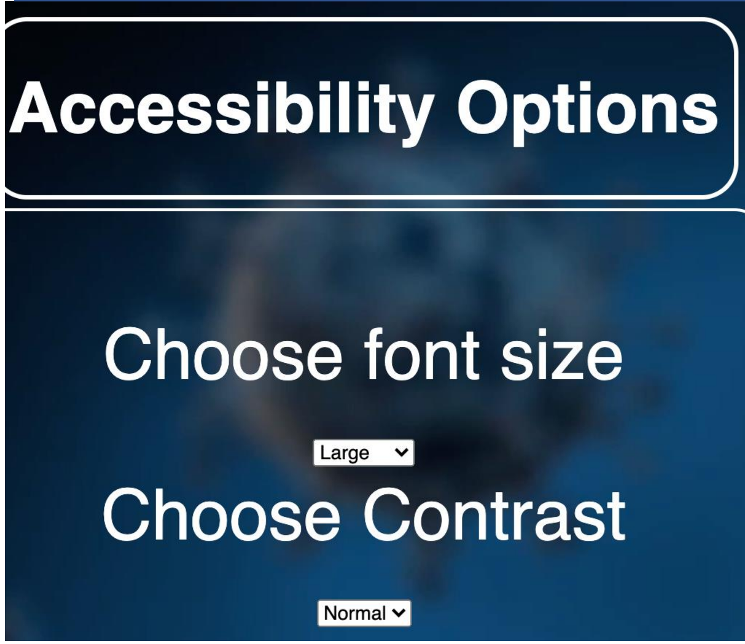
Medium Font



Accessibility Options

Choose font size Medium
Choose Contrast Normal

Large Font



Accessibility Options

Choose font size

Large

Choose Contrast

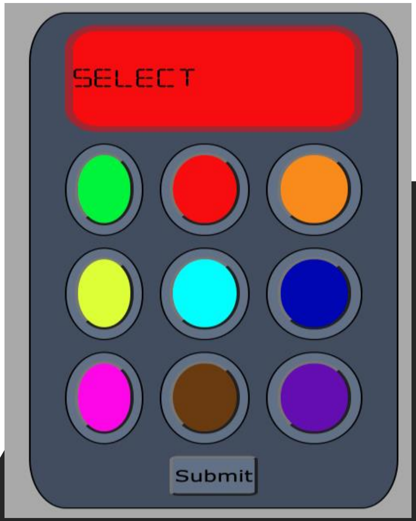
Normal

The font size is then applied to all the pages of the escape room and can only be changed by the start screen

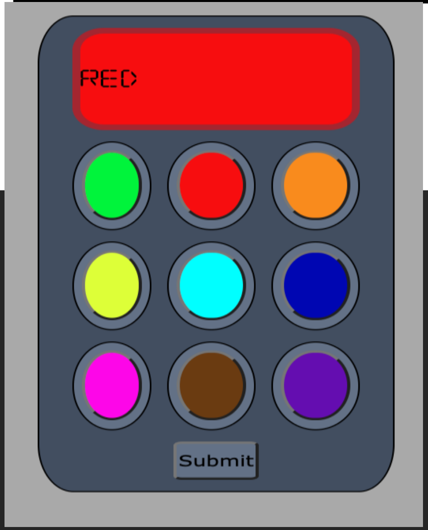
Accessibility Features

COV!D CR!S!S

Colour Pad

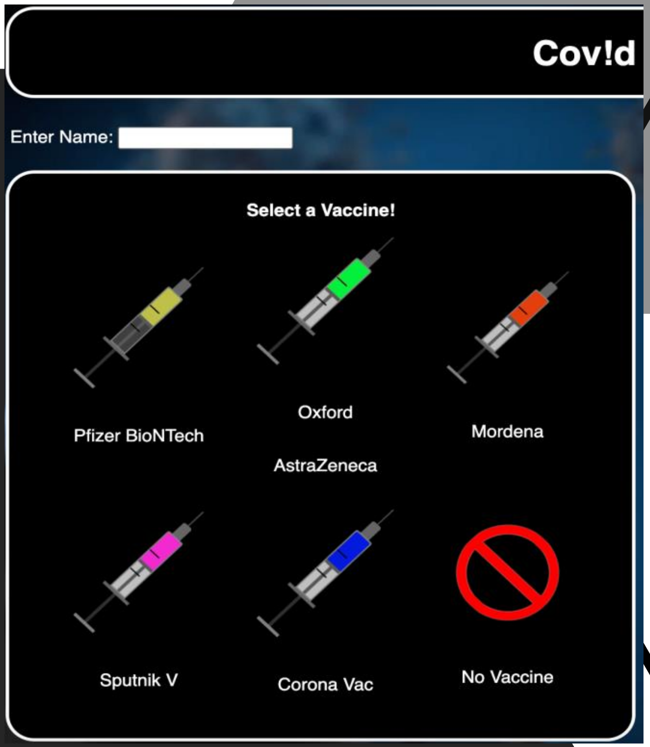


Colour Blind Pad

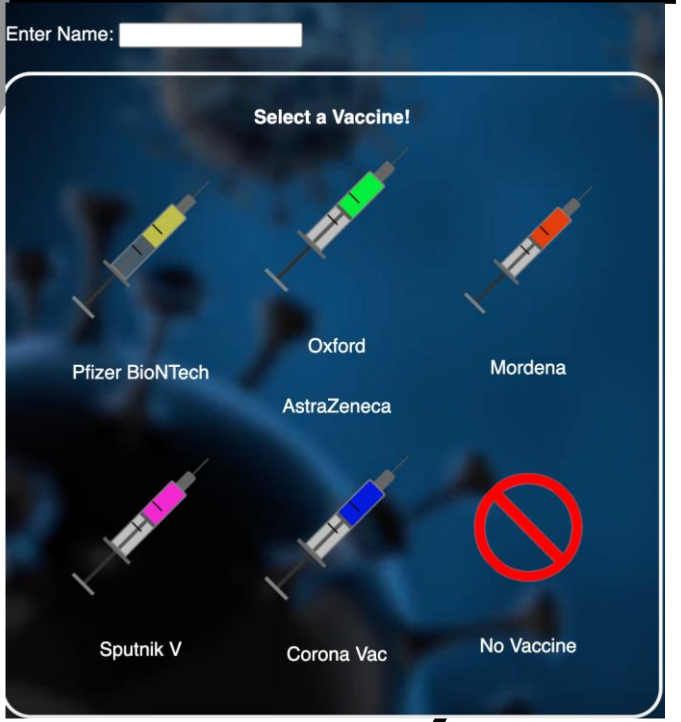


To solve the issue of the colour pad challenge for those individuals who are colour blind we decided to add a button that will change the colours on the keypad display to words so the user can still solve the hex code the find the corresponding colour (word)

High Contrast



Normal Contrast



Contrast –allowing user to change contrast means that they can choose a colour scheme that causes less strain on their eyes, by making the colours darker it means the images and text are clearer as well as blocking out the video background behind images to make them more readable

COV!D CR!S!S

Image Sources

All image sources can be found in the Image_Source_Group10.docx Microsoft Word Document

As we had too many images, videos and audio to references to fit into 2 slides

I hope you enjoyed our game just as much as we enjoyed making it

Thank you