

After pondering ideas for my final project, I have decided to create a puzzle-based game that is called "I'm not a robot". I was inspired by the "reCAPTCHA" system that enables web hosts to distinguish between human and robotic input. In a whimsical way, I imagine the "reCAPTCHA" verification process including a series of puzzle games that the system tests against the user to ensure they are real. To remain with the theme, I have decided to implement various interactivities within the puzzles that create a sense of direct human connection. For instance, the project will feature mouse input, keyboard input, audio input, and video input. It is important to note that a single user mistake will lock the simulation and put the user back to the start... it's a security measure.

The first state of the project solely includes a heading and a mousePress input field. The idea is to replicate the "reCAPTCHA" interface. The user should instinctively click on button below "I'm not a robot" and be surprised that an "ERROR" will appear. Various background patterned animations will appear for a set amount of time until the state automatically changes. A subsequent state will appear with an explanation of what is happening. Once again, a few seconds will pass and the state will automatically change to the landing page, where additional information can be found such as game instructions / controls. Once the user reviews said content, they will officially start the simulation with a keypress. The user will then be guided to the first of three (3) puzzles where they must successfully pass each objective.

The first puzzle, also known as the "audioPuzzle" will take advantage of the audioIn class. The premise of this puzzle is that the user must navigate a vehicle from the bottom of the canvas towards the top of the canvass. The gimmick is that car is controlled by the user's voice. The louder they speak, the faster the car moves along the y-axis. If they speak quietly, the car moves slowly, and if they don't speak at all, the car starts to reverse back to its starting defined y position. This on its own would be too straightforward, so I will add road obstacles such as potholes. However, unlike regular potholes, these potholes will translate upon the x-axis. The road will start to deteriorate rather quickly, so the user will have to commit to their action. We all know the severity that a single pothole can do, so it will be necessary to avoid them at all costs.

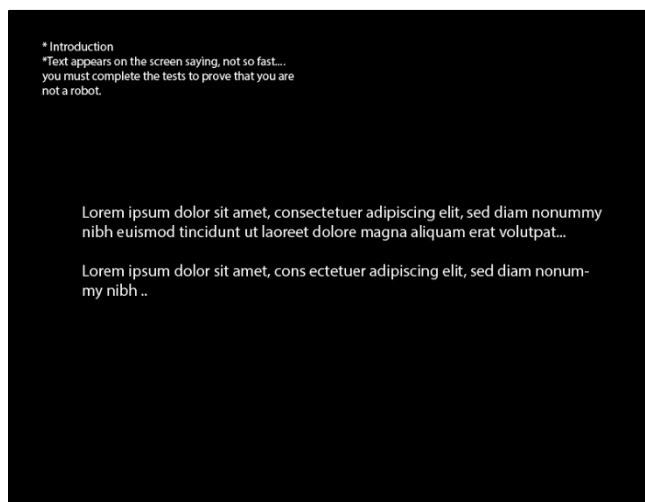
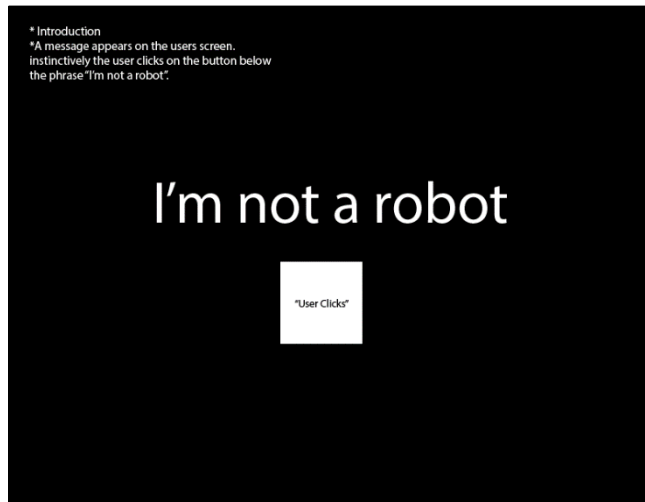
The second puzzle, also known as the "ballPuzzle", will take advantage of keyboard input. In a traditional, non-traditional way, a set number of balls will fall from the top of the canvas. The balls will not be constrained to the canvas, and so they will bounce anywhere. The user will need to bounce the balls upon an apparatus to collect jittering tokens. The objective will be to collect all the tokens within a set amount of time, and without running out of balls. One additional ball can be added as a backup; however, this will be an easter egg, and not mentioned anywhere. A visual timestamp will not be displayed on screen. Rather, I will use an osculator to create an auditory sense of remaining time. As the clock ticks down, the number of oscillations will exponentially increase to create an auditory sense of urgency. A robot would have a difficult time interpreting this... I also plan to use a PolySynth with a musical scale. Every time a ball touches the apparatus, a random note will play with a wet reverberation. My intentions are to make the sounds stress inducing, which will create an additional layer of difficulty to the puzzle.

The third puzzle, also known as the “cameraPuzzle” is less of a puzzle and more of a verification. The state will begin with live animated onscreen text. Essentially, the simulation will applaud the user for getting this far and ask them to answer a riddle. The riddle will have missing word. With the use of the keyboard, the user will have one chance to enter the correct word: “camera”. At this point the state will quickly change to white to simulate a camera flash. Once this is complete, the webcam will turn on and take the entirety of the canvas. Horizontal and vertical bars will animate on each axis, while a face detection scan is underway. A conditional statement will be used to verify if the object is in fact human. If the scan is unsuccessful, the simulation will enter the “loser” state. If successful, the simulation will lead to a state where once again a fictional “reCAPTCHA” interface appears. However, when the user clicks on the “I’m not a robot” button, a final state will appear that says, “You’re not a robot”. Simultaneously, a special recorded jingle will play to congratulate the user.

Regarding artistic vision, I am inspired by themes found in typical “spy” settings and “escape room” environments. The user is essentially taking on a mission to prove oneself that they can succeed. The James bond quote, “Your mission Jim, should you choose to accept it” comes to mind when I think of the project’s objective. Therefore, my colour palettes will contain simple RGB values with black and white. I’m attempting to simulate a computer theme that one may find in a classic movie. The font(s) will represent a retro technological age to help merge the theme together.

As I currently write this document, my largest concern / potential challenge will be the incorporation of the webcam. After briefly speaking with you, I have decided to use the ml.js library to implement face detection. To complete this process, I plan to jump ahead and watch your recorded content on AI from the CART263 class. I am hoping that once I thoroughly review the content of ml5.js and machine learning, that I should be able to implement the camera as intended. Undoubtedly other technical challenges will arise and require specific attention. I will use resources such as your recorded learning materials, <https://p5js.org/>, our discord “help” channel, and alternatives internet tutorials from YouTube channels such as “[The Coding Train](#)” to hopefully assist me with this project. I will try not to send you too many questions!

Below you will find twelve (12) chronological ordered annotated sketches that depict the simulation from start to end.



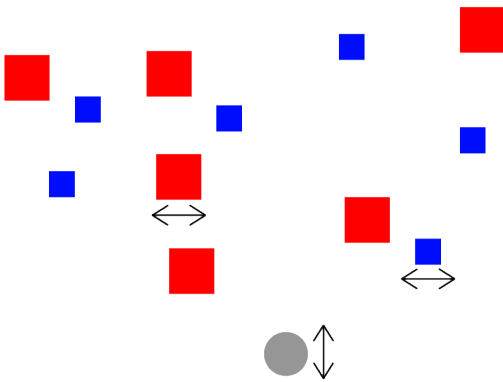
*Landing screen instructions

I'm not a robot

Press "Enter" to start

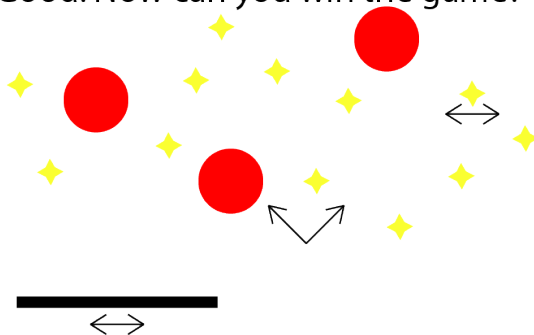
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Puzzle 1:
Potholes translate on the x-axis, while the user controls the car on the y-axis with the use of their voice.



*Puzzle 2
A game will appear where the user must collect all of the tokens with the bouncing balls. All the tokens must be collected before the balls fall off the canvas. An additional ball can be added with a mousepress as an easter egg.

Good. Now can you win the game?



***Puzzle 3**

A riddle will appear on the screen. The user is to correctly enter the missing word. Should they enter the correct word, the state will change to a flash, and the webcam will turn on.

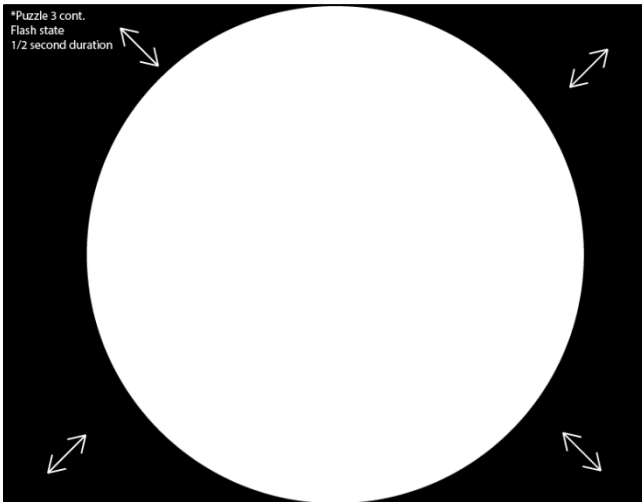
Riddle me this...

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet ***** magna aliquam erat volutpat.

Enter the missing word: "CAMERA"

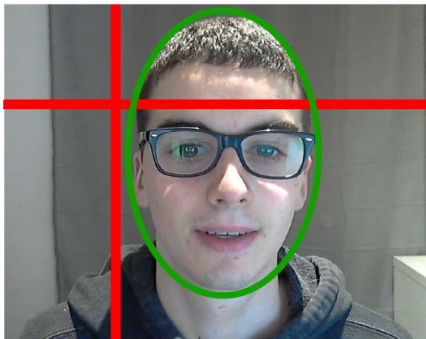
***Puzzle 3 cont.**

Flash state
1/2 second duration



***Puzzle 3 cont**

The webcam will turn on and perform a facial recognition. Red bar will translate on both axis as this recognition is being performed.



*Global element
If the user fails any task, they will be directed to the "loser state". This applies to all 3 puzzles.

For security reasons, you must re-start the simulation

*Puzzle 3 cont.
The "I'm not a robot" button makes a return, if the facial recognition system comes back as successful. Other wise the user gets presented with global "Failure" screen.

I'm not a robot

"User Clicks"

*Puzzle 3 cont.
Final state that validates that you are in fact not a robot. Jingle plays in the background!
The text may jump around or become the users cursor.

You're not a robot