After pondering ideas for my final project, I have decided to create a trivia game 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 verification process including a series of puzzle games that the system tests against the user to ensure they are real. The puzzles will include a series of objectives that the user must successfully answer within the depicted time limit. Therefore, at this stage, the game is to be described as follows…

As the user starts the simulation a small input field, that looks like a “reCAPTCHA” window will appear in the middle of the screen. The user will instinctively click on I’m not a robot, and an “ERROR” will appear, with patterned virus bugs appearing all over the x and y-axis. After a few seconds the state will automatically change to the title / instructions and provide the user with further explanation as to what they just experienced, and what they must do to prove that they are not a robot. Once the user reviews the instructions, the simulation will officially start, and the user will then be guided to the first of 3 states where they must successfully pass each objective. A single failure will lock the simulation and put the user back to the start… it’s a security measure.

In first puzzle, in state 1, will be in the form of Question and answer. question will be asked, and the user will have 4 answers to choose from. A timer will count down from 10 and be displayed in the background with an opacity. The idea is to add stress to the user. A song, likely the jeopardy jingle will play as the user thinks about the answer. Keyboard input will allow for the user to select their answer. Keys A, B, C, D. The final version, will have various questions and potential answers within an array that will be selected at random, so should a user play the game more then once, they don’t know the question right away.

The second puzzle, in state 2, will be a bouncing ball simulation where the user must collect all the tokens, without lousing the bouncing balls. Several tokens will appear on the canvas. They will jitter along the x-axis, while numerous balls fall from the top of the canvas. The objective of the user is to collect all the tokens before the balls bounce off the canvass. Only a human can decipher this puzzle, as the movement of the balls are random and cannot be anticipated.

The third puzzle, in state 3, is less of a puzzle and more of a verification. With the use of the user’s webcam, the system will scan the users face and will only proceed if it can be matched with one of a human. Horizontal and vertical bars will animate on each axis, while the scan is being done. If the scan is unsuccessful, like in a sense that no webcam is detected, or that the image in front of the camera is not human, the simulation will fail. If the scan is successful, the simulation will lead to its’ final stage, that appears once again the “reCAPTCHA” looking window, however this time when the user clicks on the “I’m not a robot”, a final state will appear that says something along the lines of “you are not a robot”.

The largest challenge with this conception is the use of the webcam. After briefly speaking with you, I have decided to use the ml.js library to implement face detection. Once I implement the initial code and have the webcam working correctly, I look forward to learning how I can take advantage of it and what I can make from it to help bring together the simulation.

The program will end once the facial recognition successfully states that the object in front of the camera is at least 85% human…. A final state will appear that simply says “Indeed, you’re not a robot”.

States with various puzzles

1. Answer the complex text question with keyboard input
2. Collect all the tokens before the balls disappear
3. Scan face with webcam to prove identity

**Notes**Prototype is just to include shapes. Don’t bother with images yet.

Various timers will be used throughout the program to assist with the story telling aspect. This allows the user to become better engaged with the program. The creation a smooth flow develops from the introduction, to the 3 “tests”, while finishing with the conclusion.