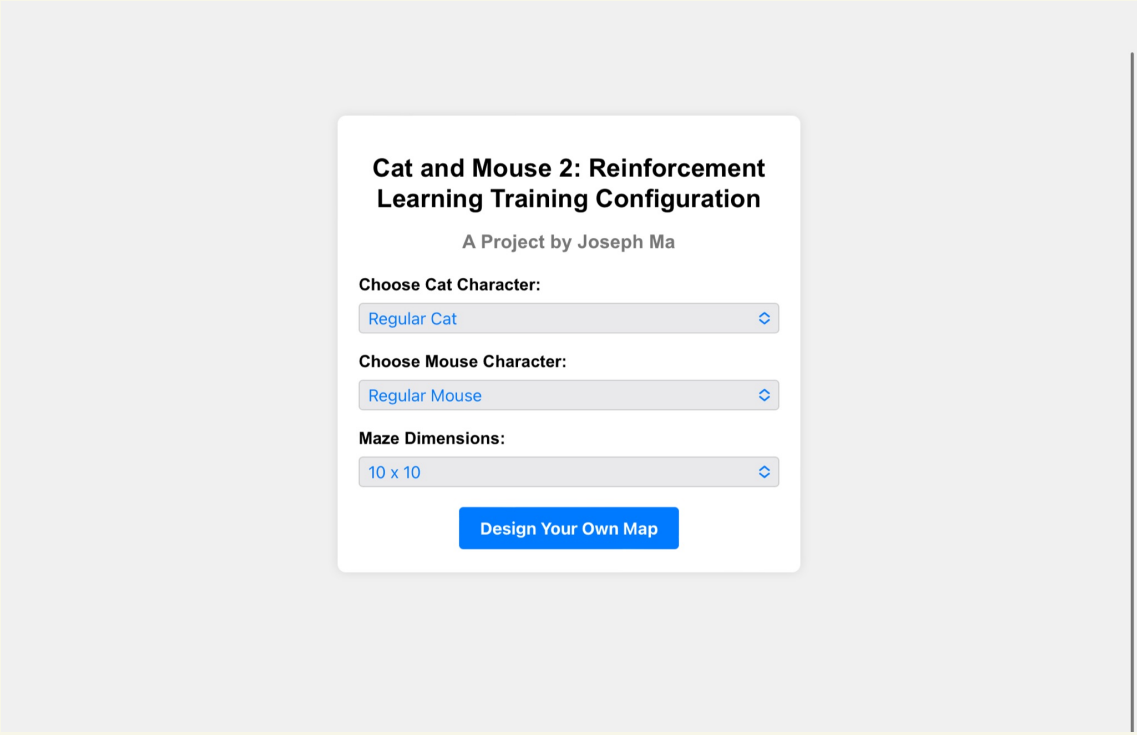
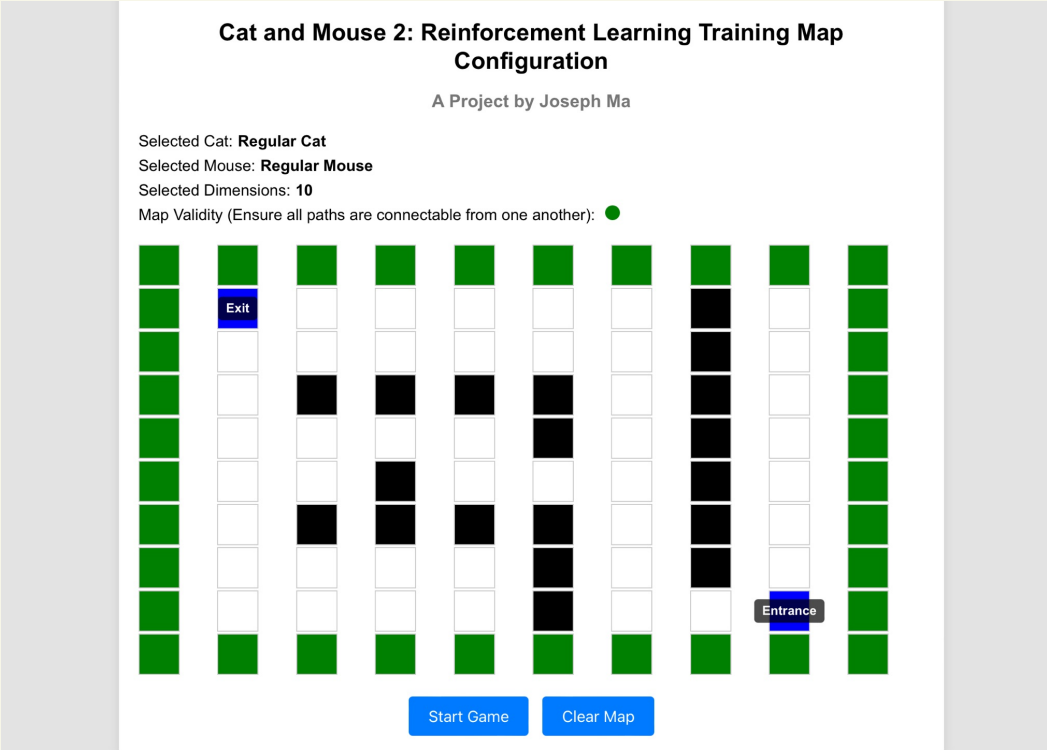
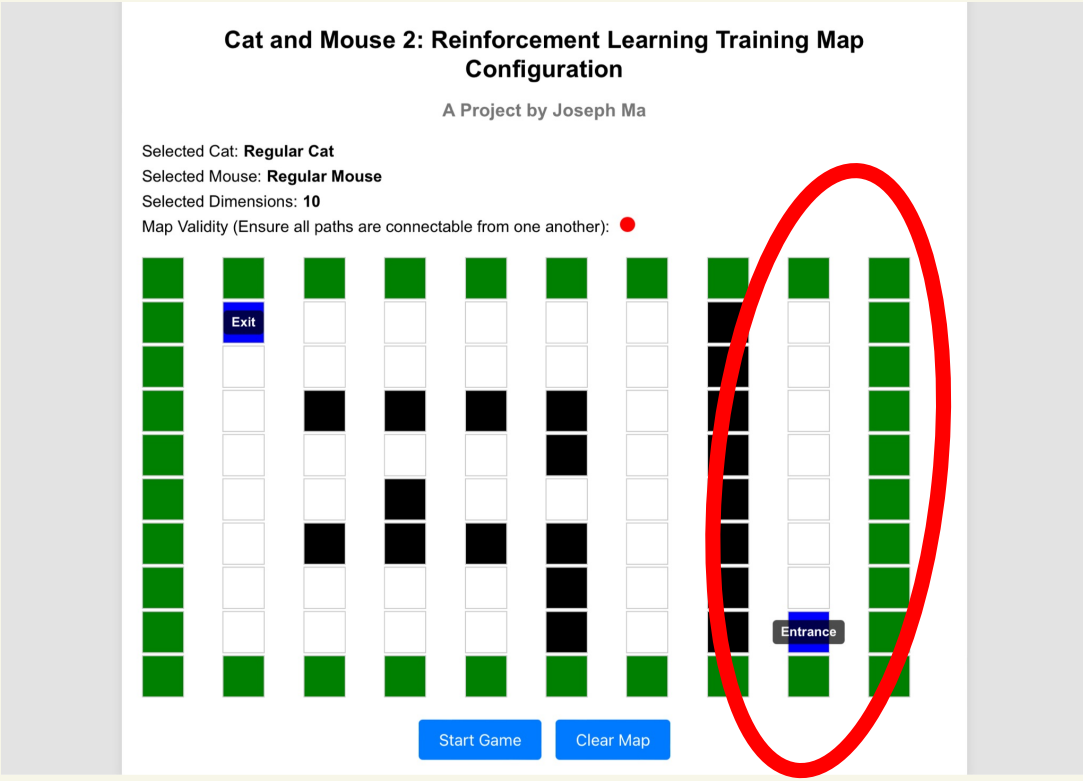
The first page allows you to select a character for training and select the map size. Note that larger map size means longer training time.



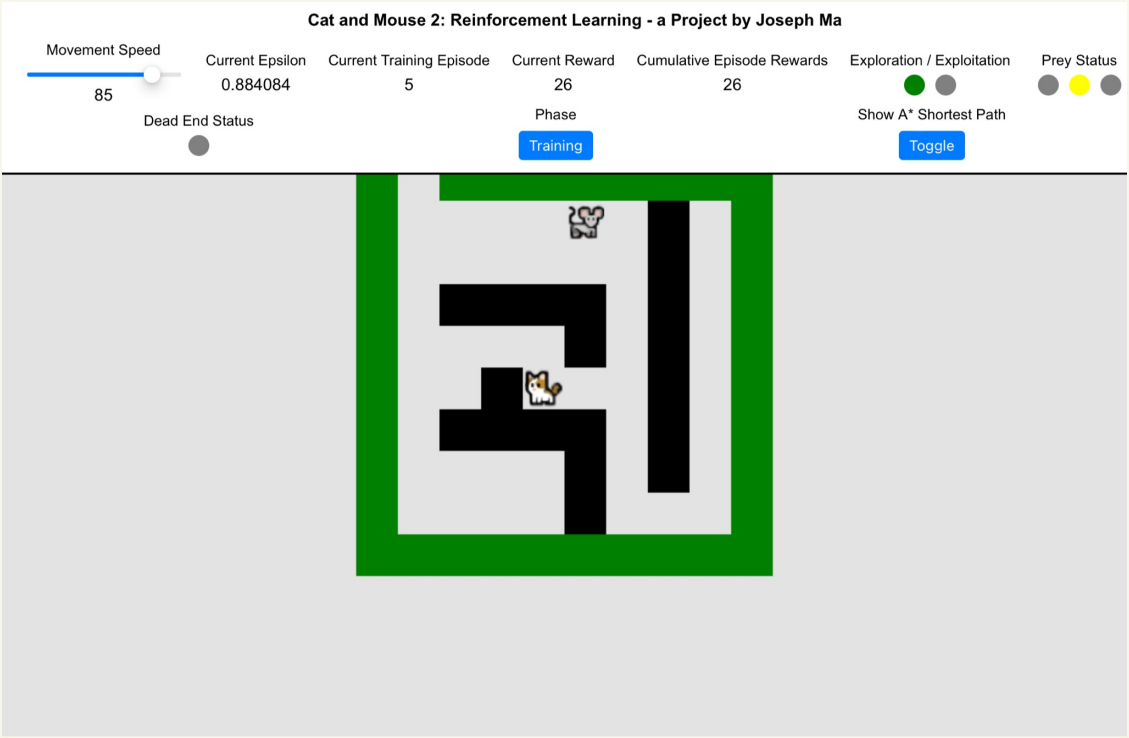
The next page allows you to design your own map for the training environment. Place walls down as you would like (in black).



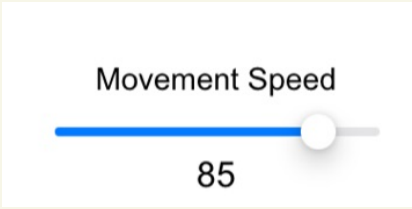
Note that if a portion of the map is disconnected from the others this will invalidate it. Ensure all paths are connectable from one another. There should also be at least one path from entrance to exit.



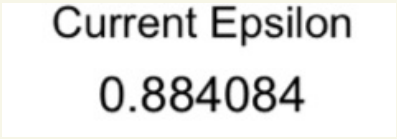
Let’s begin training



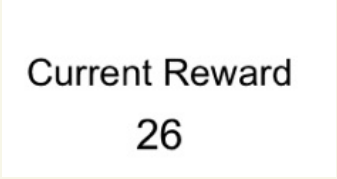
Adjust the speed at which the episodes play out



We use the greedy epsilon strategy, this is the chance the mouse chooses the best action to take. And 1 - epsilon the chance the mouse chooses a random action.



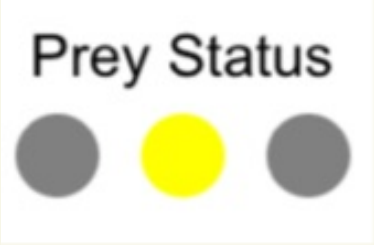
The current reward is a reflection of how “good” the mouse’s action is. The more positive the better, and the more negative the worse.



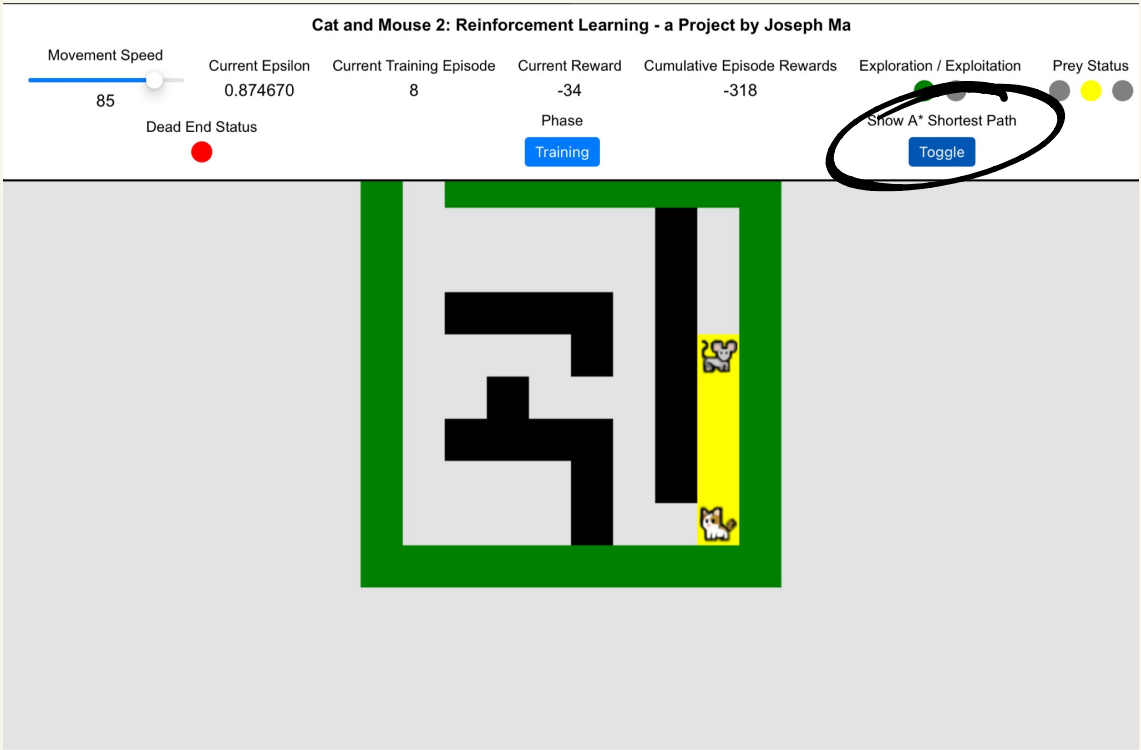
The total rewards accumulated through the current episode.

Indicates whether the most recent action by the mouse was exploration or exploitation

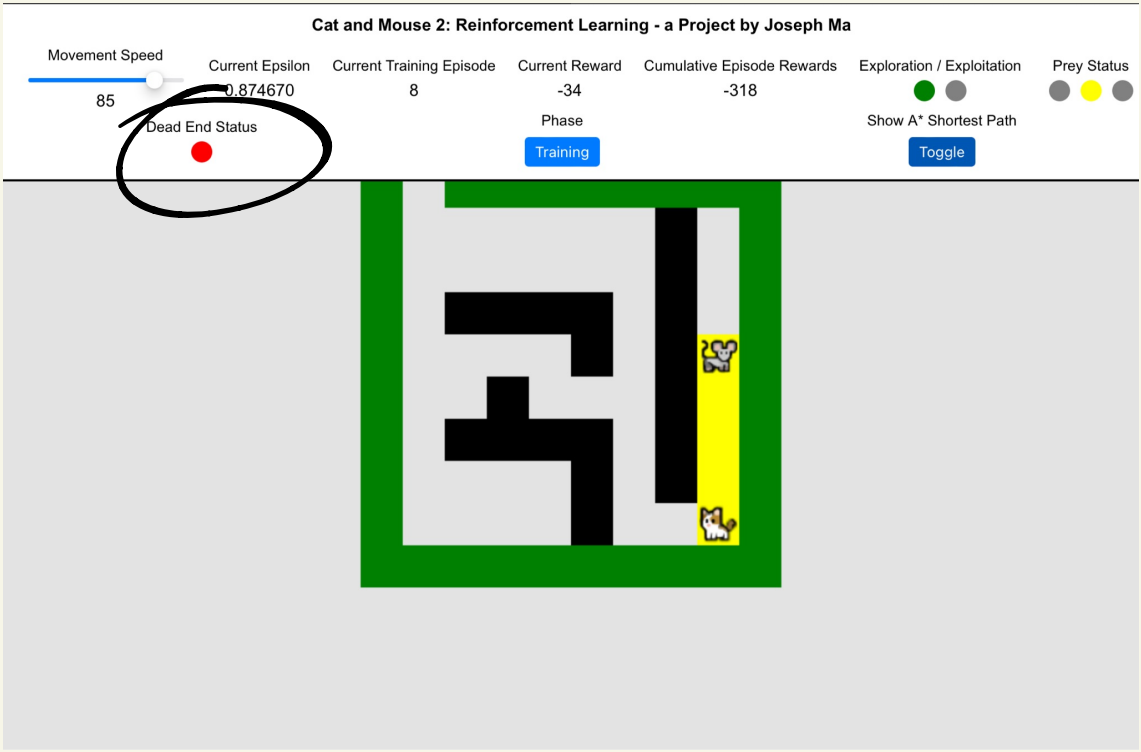
Prey status: Green will light up at the most left if mouse escapes, Yellow will light up in the middle of mouse is still alive, Red will light up at the most right if mouse gets caught by the cat



Click toggle to show the A* shortest path calculated by the cat to reach the mouse. The cat always takes the shortest path to the mouse.



If the mouse finds itself cornered by the cat, the dead end status will light up. A double DFS algorithm was written to check for this.



Once epsilon has decayed, toggle the training button. This will cause the mouse to take the best action in each move. In other words it will exploit every action. In this mode, the Qtable and epsilon will remain frozen until we return to regular training mode. This mode is to visualize how intelligent the mouse has gotten.