

SELF LEARNING CARS AI

By David Tang, Youssef Ben Mouny and Elyes Bradai



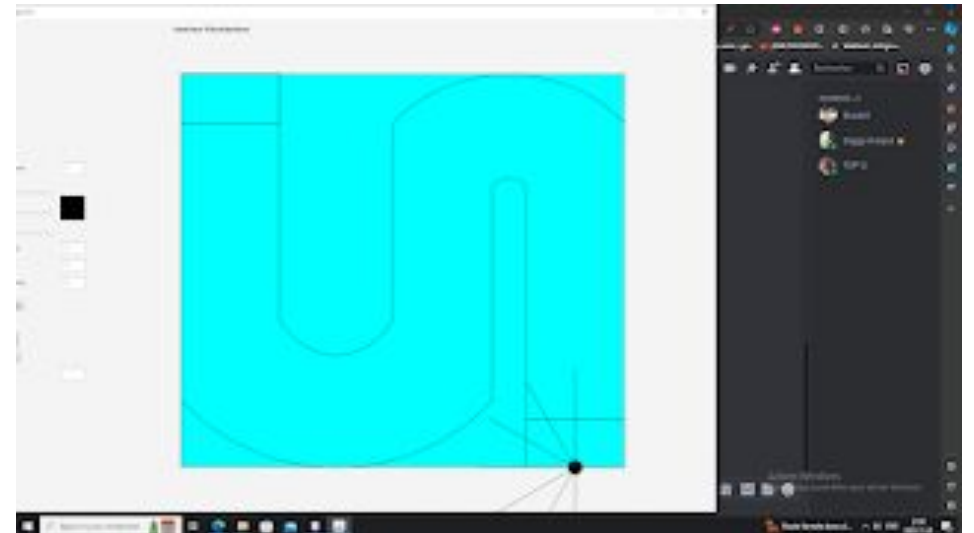
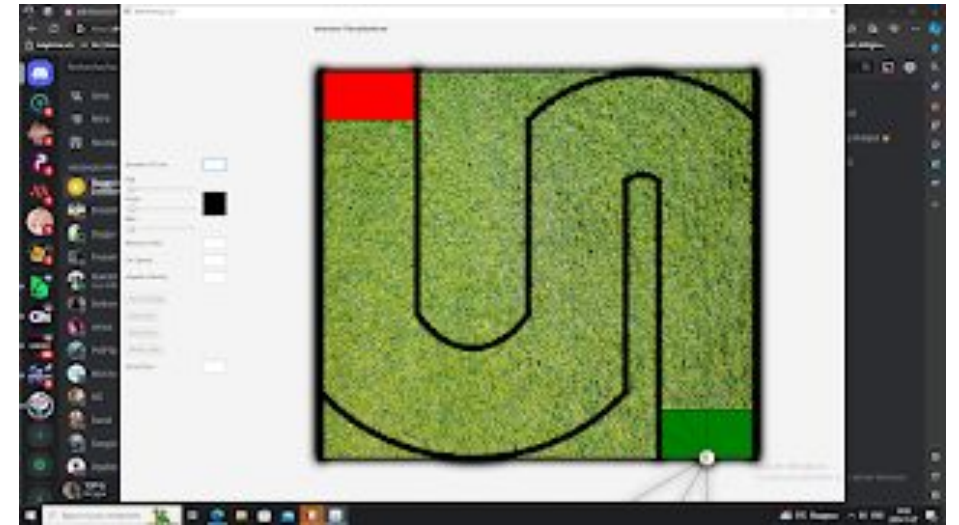
GOAL OF THE PROJECT

- Explore javafx's user interface
- Use knowledge from other scientific classes
- Give the user multiple input choices
- Personal goal: learn about neural networking

HOW IT STARTED

Early designs :

“The AI does not hate you, nor does it love you, but you are made out of atoms which it can use for something else.” - Eliezer Yudkowsky





FINAL VERSION



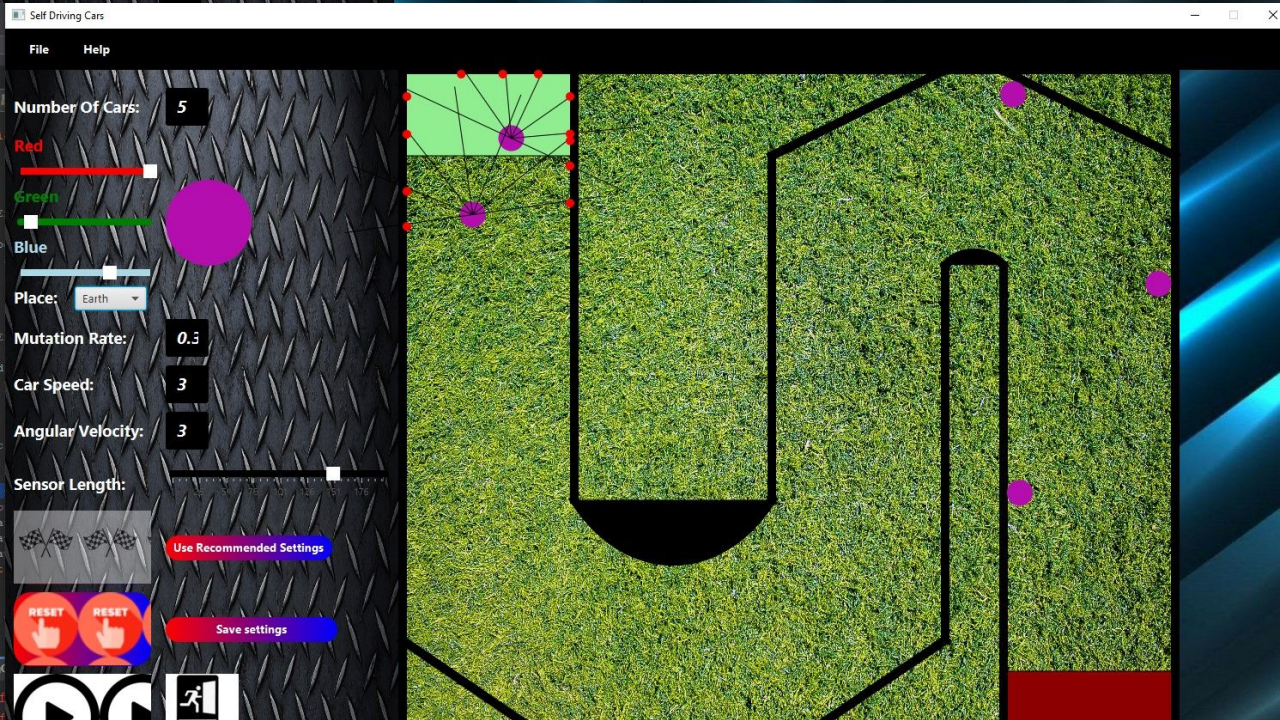
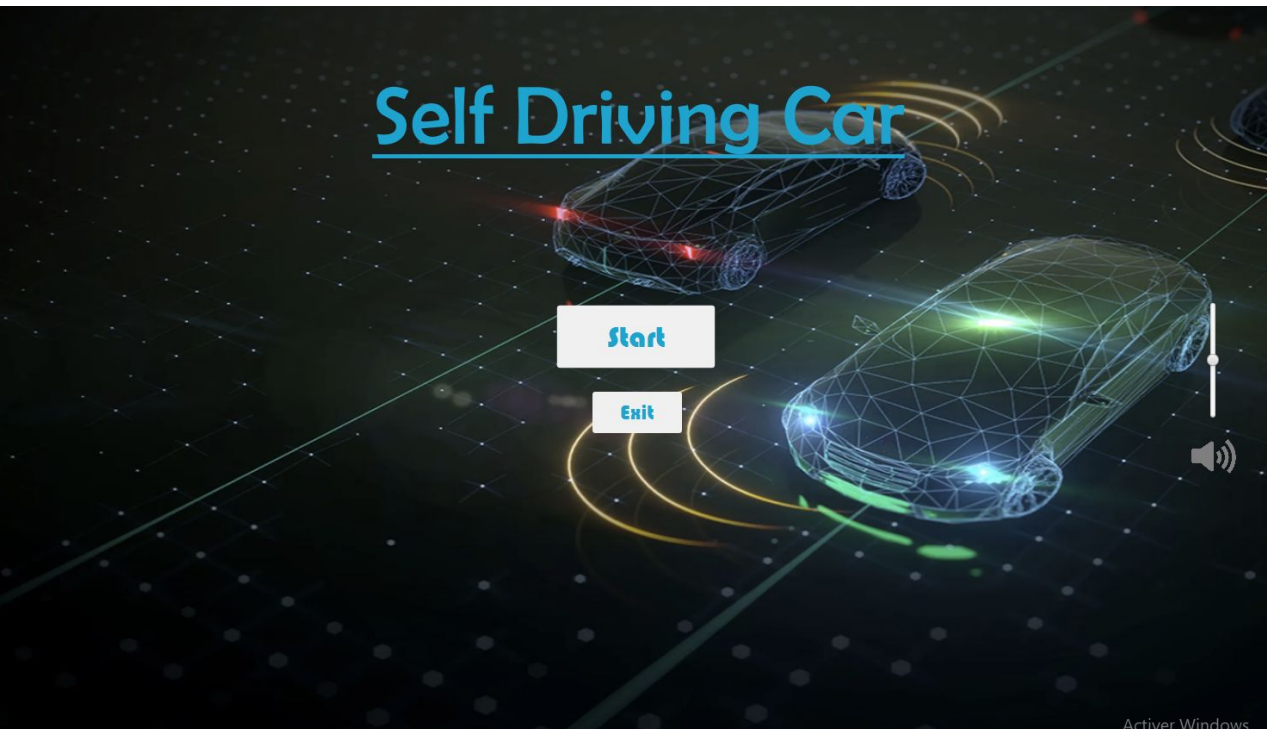
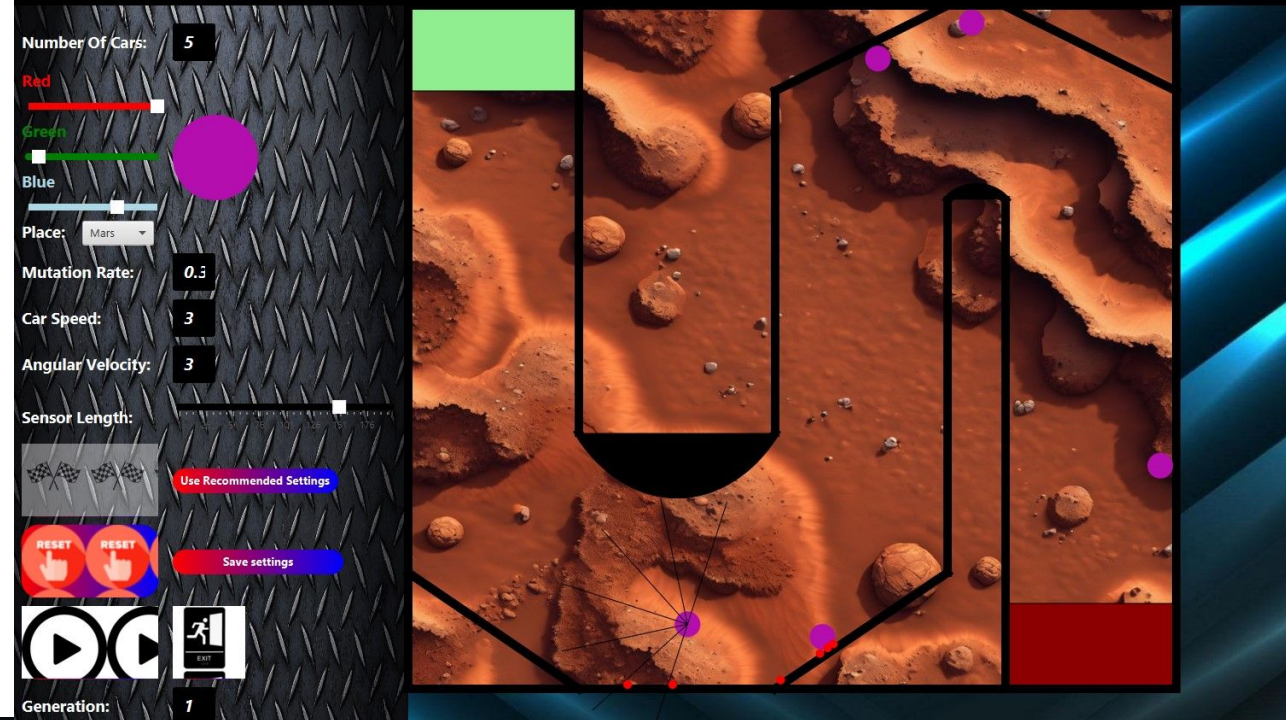
Description of
the layout

Description of
how inputs
affect the
program

Brief
explanation of
the neural
network

How to use the
program

LAYOUT

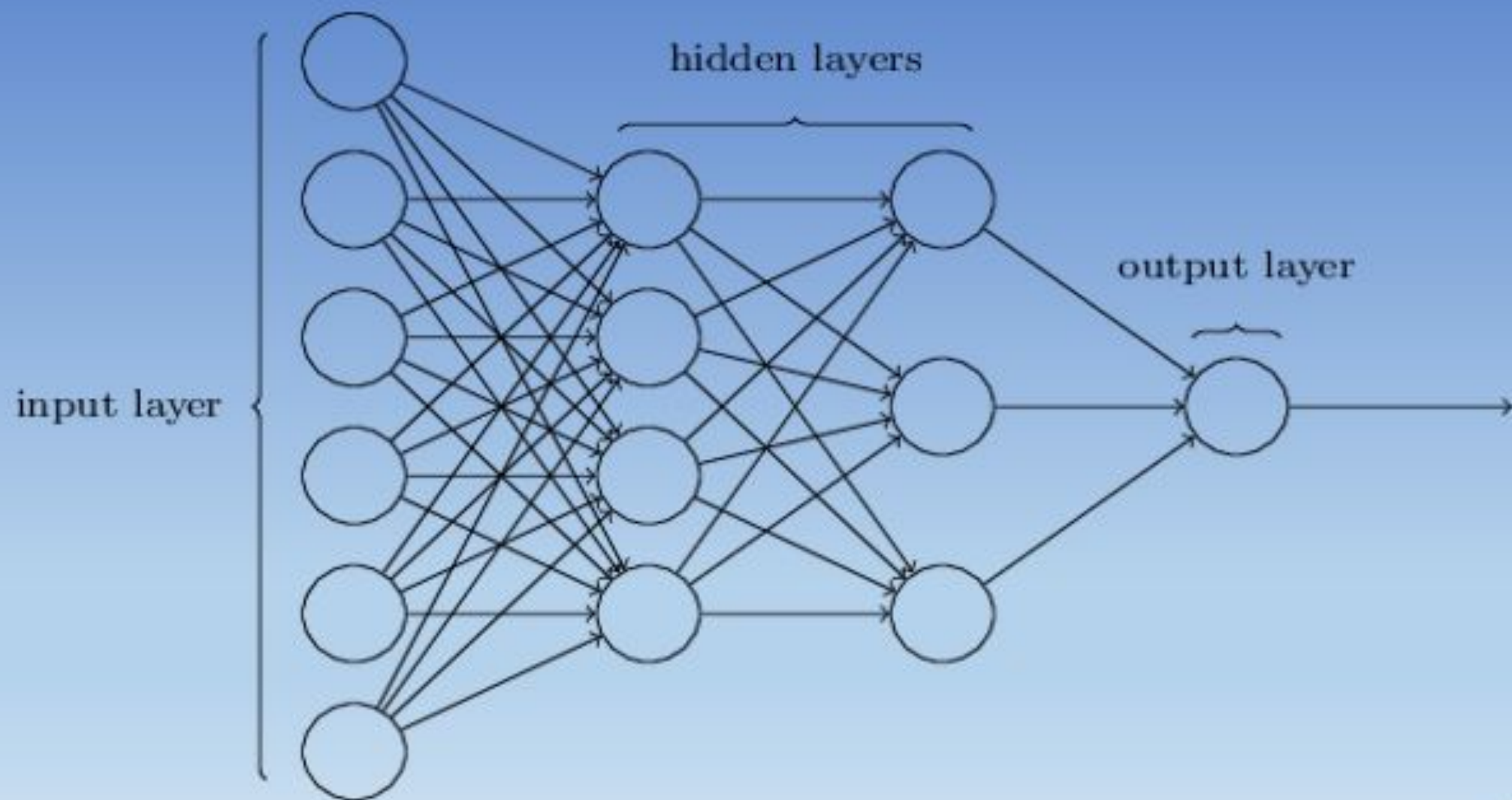


USER INPUTS AND HOW TO USE THE PROGRAM

The screenshot shows a settings menu for a program. The background is a dark, metallic, diamond-plate texture. On the right side, there is a large black circle. The settings are listed on the left, with corresponding input fields or sliders on the right.

- Number Of Cars:** A black box containing the number **1**.
- Red:** A red horizontal slider bar.
- Green:** A green horizontal slider bar.
- Blue:** A blue horizontal slider bar.
- Place:** A dropdown menu showing **Earth**.
- Mutation Rate:** A black square input field.
- Car Speed:** A black square input field.
- Angular Velocity:** A black square input field.
- Sensor Length:** A horizontal slider bar with numerical markers from 34 to 176. The slider is positioned at approximately 151.
- Buttons:** A row of four buttons: two with checkered flags, a red **Use Recommended Settings** button, and a purple **Save settings** button.
- RESET:** Two red circular buttons with white thumbs-up icons and the word **RESET**.
- Navigation:** Two large circular buttons with play and stop icons, and a black **EXIT** button with a white icon of a person running.
- Generation:** A black box containing the number **0**.
- Neurons per Layer:** A black rectangular input field.

Classification of Neural Networks



DEMO OF THE PROGRAM

```
object to mirror  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select = 1  
context.scene.objects.active  
("Select the modifier")  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

-- OPERATOR CLASSES -----

```
bpy.types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```


QUESTIONS?



A group of people are shown from the chest up, clapping their hands. The image is slightly blurred, focusing on the hands in the foreground. The text "THANK YOU FOR LISTENING" is overlaid in the center in a bold, white, sans-serif font. The background is a soft, out-of-focus indoor setting.

**THANK YOU FOR
LISTENING**