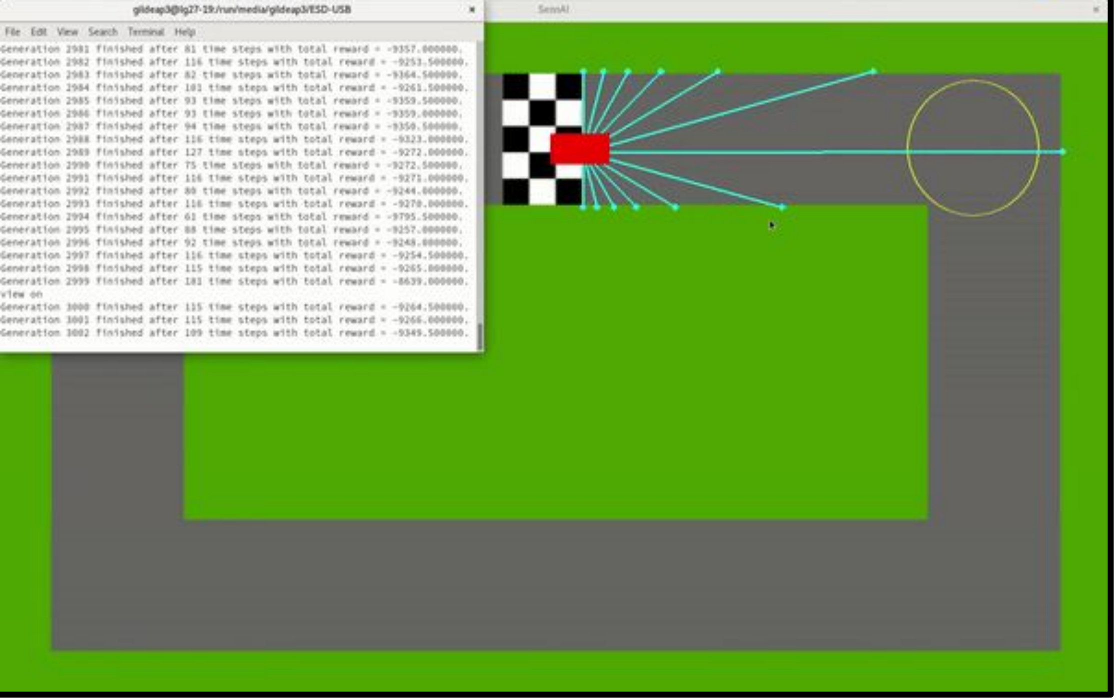
SennAI - Demonstration

CA326 Project Demonstration by Connell Kelly & Patrick Gildea

Our Aims

- Successfully demonstrate reinforcement learning with a python based program.
- Develop a working gamification of reinforcement learning with a competitive racing video game.
- Record data and show the apparent progress of an AI agent as it completes a task to the best of its ability through trial and error.

As seen in our functional spec, above is the original visualised concept of our project in action. Along with our high priority aims, there were several ‘wishlist’ features we hoped to implement such as an option to speedup AI agent training and loading screens for loading navigation data. While several proposed ideas would change in development, most would still be fulfilled.



gildeap3@lg27-19:/run/media/gildeap3/ESD-USB
File Edit View Search Terminal Help
SennAI

```
Generation 2981 finished after 81 time steps with total reward = -9357.000000.
Generation 2982 finished after 116 time steps with total reward = -9253.500000.
Generation 2983 finished after 82 time steps with total reward = -9364.500000.
Generation 2984 finished after 101 time steps with total reward = -9261.500000.
Generation 2985 finished after 93 time steps with total reward = -9359.500000.
Generation 2986 finished after 93 time steps with total reward = -9359.000000.
Generation 2987 finished after 94 time steps with total reward = -9350.500000.
Generation 2988 finished after 116 time steps with total reward = -9323.000000.
Generation 2989 finished after 127 time steps with total reward = -9272.000000.
Generation 2990 finished after 75 time steps with total reward = -9272.500000.
Generation 2991 finished after 116 time steps with total reward = -9271.000000.
Generation 2992 finished after 80 time steps with total reward = -9244.000000.
Generation 2993 finished after 116 time steps with total reward = -9270.000000.
Generation 2994 finished after 61 time steps with total reward = -9795.500000.
Generation 2995 finished after 88 time steps with total reward = -9257.000000.
Generation 2996 finished after 92 time steps with total reward = -9248.000000.
Generation 2997 finished after 116 time steps with total reward = -9254.500000.
Generation 2998 finished after 115 time steps with total reward = -9265.000000.
Generation 2999 finished after 181 time steps with total reward = -8639.000000.
view on
Generation 3000 finished after 115 time steps with total reward = -9264.500000.
Generation 3001 finished after 115 time steps with total reward = -9266.000000.
Generation 3002 finished after 109 time steps with total reward = -9349.500000.
```

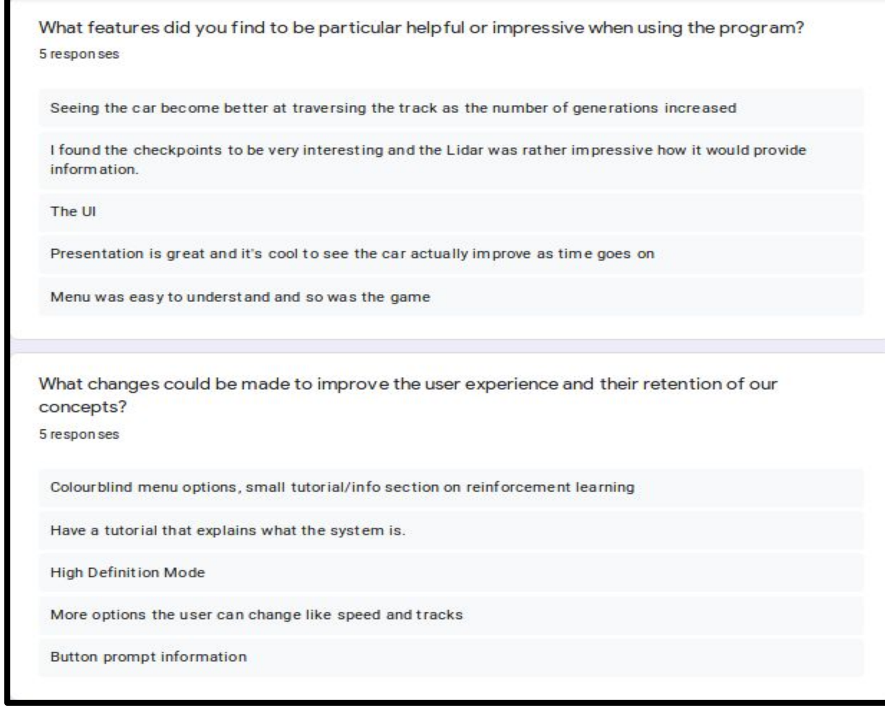
Our Achievements

- Successfully developed a working reinforcement learning-based environment for an AI agent, running on a Q-learning algorithm.
- Created a tangible competitive racing game to allow for user interaction with SennAI.
- Developed a process to record progress and developed a better understanding of the program.

Above is a short showing of our AI agent, SennAI, making some progress around a racetrack after 3,000 generations. It works by using a working lidar system and appropriate checkpoints to guide the learning process. With every generation building onto it’s Q-table of navigational data, SennAI can complete the track at an optimised pace.

Unfulfilled Ambitions

- Implement multiple tracks with appropriate .npy Q-table data and the disc space to support it.
- Provide a thorough ‘Options’ mode with toggles that enable visualization of more generations at once, lidar visuals and car speed alteration.
- Compile the project code into an executable file instead of a python file run from terminal.
- Introduce collision detection for the user car in the case that SennAI collides with them. This would allow SennAI to actively avoid the user.



What features did you find to be particular helpful or impressive when using the program?

5 responses

Seeing the car become better at traversing the track as the number of generations increased

I found the checkpoints to be very interesting and the Lidar was rather impressive how it would provide information.

The UI

Presentation is great and it's cool to see the car actually improve as time goes on

Menu was easy to understand and so was the game

What changes could be made to improve the user experience and their retention of our concepts?

5 responses

Colourblind menu options, small tutorial/info section on reinforcement learning

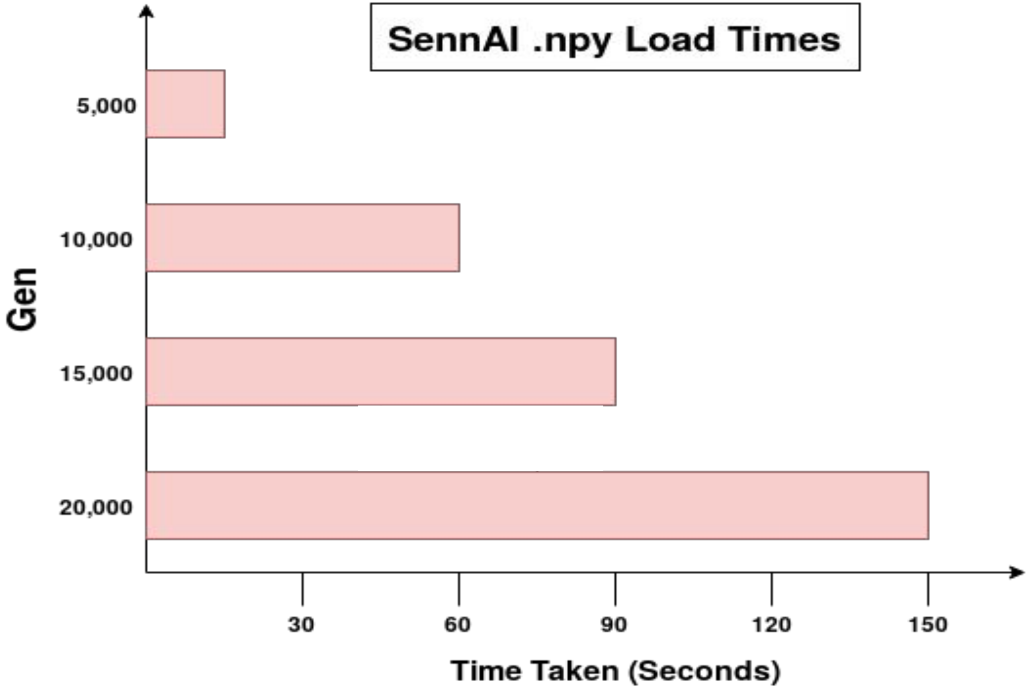
Have a tutorial that explains what the system is.

High Definition Mode

More options the user can change like speed and tracks

Button prompt information

Throughout the project’s development, there were many concepts and ideas we had hoped to implement into our program, but couldn’t due to time, experience or disc space limitations. As a result of said disc space limitations, our AI agent’s progress isn’t as pronounced as it could be due to the exponentially increasing size of .npy Q-tables as they retain new data.



What Was Learned?

- Understanding the fundamental concepts behind reinforcement learning and how to implement it into code.
- Gained a grasp on the reinforcement learning algorithm, Q-learning and it's associated formula.
- Learned about features to implement from valuable user test feedback.
- Learned many new useful functions from external sources like PyGame.

We came into this project with propositions we had a curious understanding of. We depart it with a much stronger understanding of one of the keystones of machine learning and one of it's most useful algorithms. The skills that we learned during this project will stand by us in future projects. Like SennAI, the trial and error here will help us succeed better in the future.

SennAI

Thank you for your time!

We're happy to answer any questions you might have.