



An Amazing Maze game created in Ruby

Walk-Through

After opening the app in your terminal of choice, the game will begin to load a Welcome splash screen.

Following the Welcome message, the Game will display your objective.

After a few seconds the objective message will disappear and the game will begin. As the objective states, you should use the arrow keys to guide the car through the maze to reach your puppy.

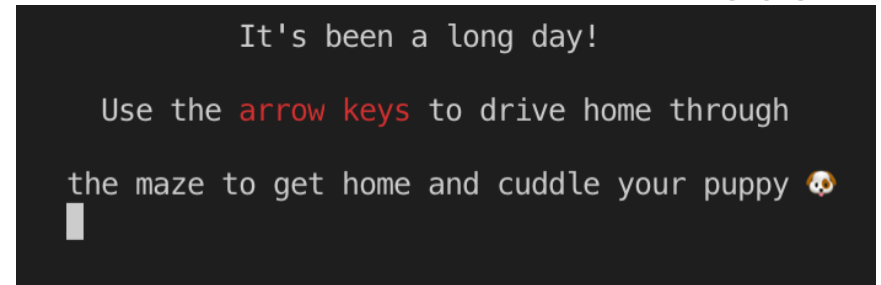
Upon successfully reaching your puppy, the game will end and a congratulatory message will be displayed.

If you quit by pressing 'q' or 'esc' on your keyboard at any point then the game will be over.

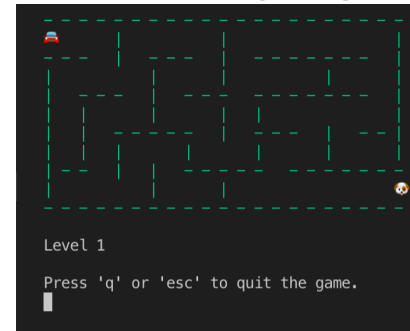
WELCOME MESSAGE



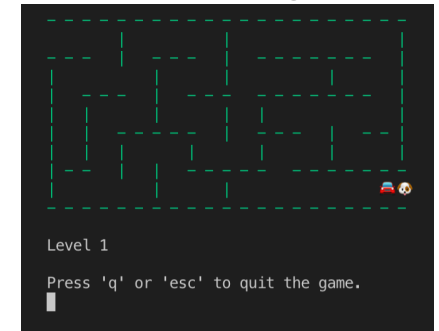
OBJECTIVE



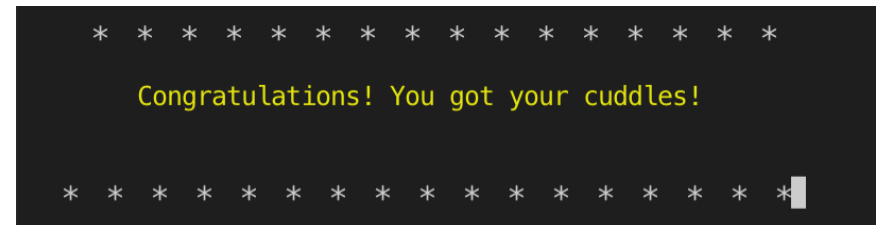
GAME START



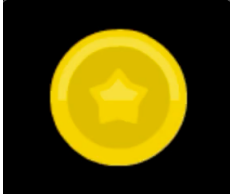
GAME END



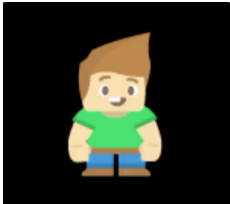
CONGRATS MESSAGE



COIN (VIDEO)



HERO (VIDEO)



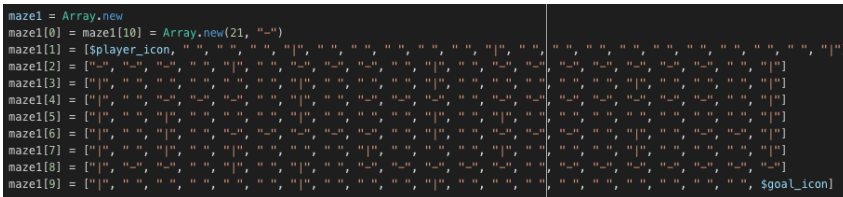
PUPPY EMOTE



CAR EMOTE



HARD CODED MAZE



GIT



Review

Challenges

Our initial idea was to create a 2d platform game using the Gosu & rub2d Gems but after difficulty of getting this to run on Jo's computer we decided to use ASCII art & Emojis as we didn't want create a game that would not risk not running on someone else's machine.

After deciding on creating a maze for our project, we looked into dynamically creating random mazes each time the game was run but we were unable to successfully implement this and decided on hard coding the mazes to reduce the risk of losing time.

We also ran into the issue of increasing levels. 4 hard coded mazes were created with the expectation to move through each one after successful completion of the previous but we were unable to implement this in the time we had remaining.

One of the final challenges we faced was the importance of using branches in GIT from start to finish. Although our project used several branches from its inception, towards the end some of our final changes were pushed straight to Master causing many merge conflicts.

Ethical Issues

During the brainstorming process of deciding on what project to do, we considered the ethicalness of using an API (specifically for Google translate) as it could potentially mean Google would be doing most of our project for us. This was the main reason we decided not to go with this idea.

The objective of the game we created is to return to your home to give your puppy cuddles. There are no ethical issues here.

Favourite Parts

One favourite part is the animating congratulations message that appears when you win the game. The message uses coloured stars and the sleep method to give the effect of an animating message.

And another favourite is the ability to use emoji in command line. This is a nice little discovery as it gives some nice 2d animations without the need of third party gems.

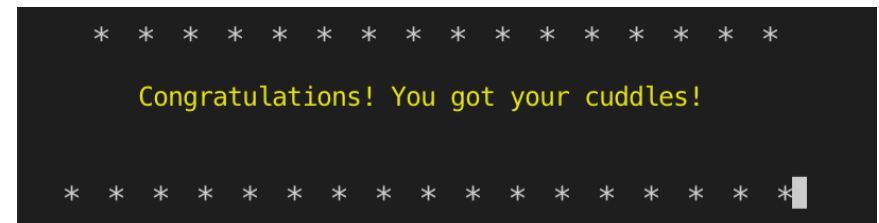
GOOGLE TRANSLATE



AMAZING MAZE



CONGRATS MESSAGE



PUPPY EMOTE



CAR EMOTE



The Future of



Future Releases

There are quite a few feature we would have loved to add to this game but we were unable to include them into this release because of time constraints.

Below are a few features we had planned to add to the game:

- A scoring system.
- Game Timer.
- Additional Levels.
- 2d Animations.
- Sound effects.
- Cookies you can collect for additional points. 🍪
- Automatic Maze creation.
- Choose your player option.
- Themed Maze for holidays throughout the year.

Questions



An Amazing Maze game created in Ruby