Open Program

Kaan G

Inhoud

Context

Brawlhalla is a fighting game wherein you can fight against each other. In my project we focus on 1v1 matches inside Brawlhalla. When you start both players start with 3 lives. If you die, you lose 1 life. If you die 3 times you lose the match. I want to create a Brawlhalla Agent that is so good it could help me train to get better.

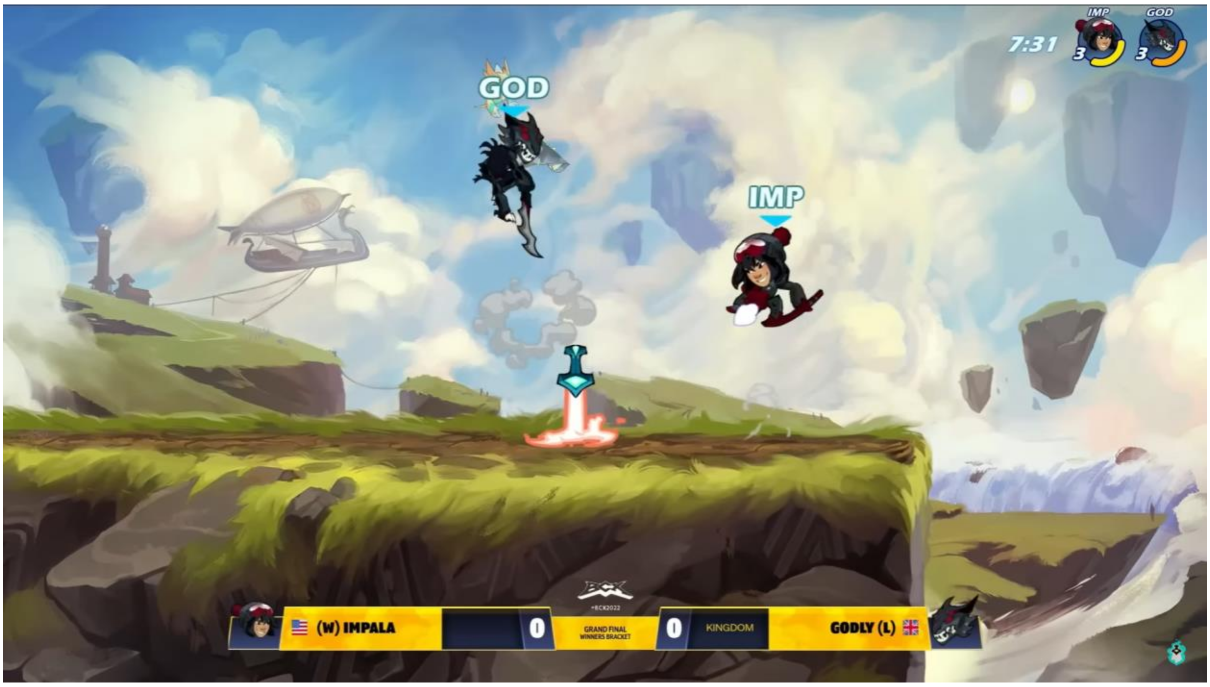


Figure 1: The Grand Final of The Brawlhalla World Championship 2022 (Godly vs Impala)

Main Goals

I have a set of goals which increase in difficulty per goal. For this project as a minimum I want to beat at least an *Easy-Bot (Goal 1)*. If I reach that goal I can try to push it further and see how far I can get.

1. Beat an *Easy-Bot* in a 1v1 match in the game Brawlhalla using AI.

2. Beat a *Medium-Bot* in a 1v1 match in the game Brawlhalla using AI

3. Beat a *Hard-Bot* in a 1v1 match in the game Brawlhalla using AI.

4. Beat an *Expert-Bot* in a 1v1 match in the game Brawlhalla using AI.

5. Beat a *Chosen-Bot* in a 1v1 match in the game Brawlhalla using AI.

6. Beat a *Low-Diamond-Player* in a 1v1 match in the game Brawlhalla using AI.

7. Beat a *High-Diamond-Player* in a 1v1 match in the game Brawlhalla using AI.

8. Beat a *Valhallan-Player* in a 1v1 match in the game Brawlhalla using AI.

9. Beat a *Semi-Professional-Player* in a 1v1 match in the game Brawlhalla using AI.

10. Beat a *Professional-Player* in a 1v1 match in the game Brawlhalla using AI.

Terminology

Do note that terms as Low Diamond and High Diamond can change in a few years.

Low Diamond Player: A player with a consistent 1v1 ranked Elo of 2000 – 2250.

High Diamond Player: A player with a consistent 1v1 ranked Elo of 2250 – Top 150. (around 2475 rn)

Valhallan Player: A player that is consistently ranked top 150 on the online 1v1 leaderboard.

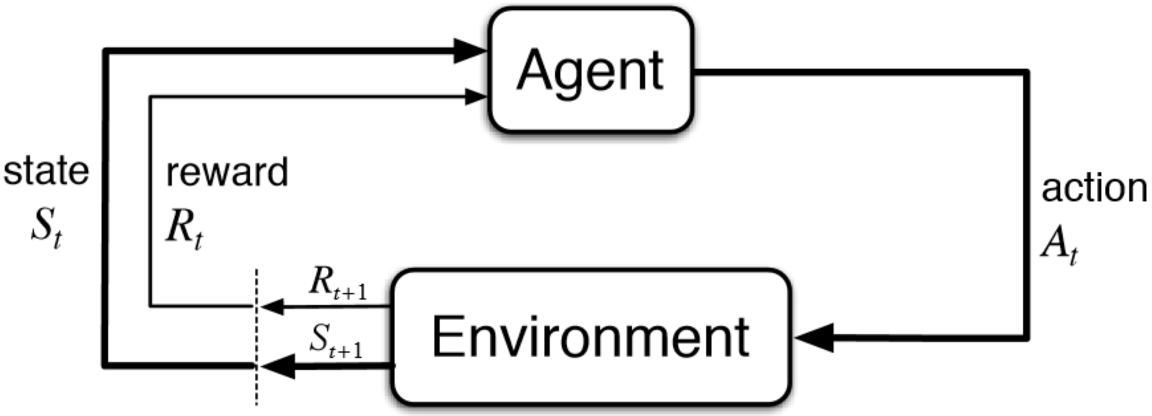
Semi Professional Player: A player which places top 32 consistently in official 1v1 tournaments.

Professional Player: A player which places top 8 consistently in official 1v1 tournaments.

As earlier said I want to beat the bots/players using AI. There are 2 main techniques I can use: Reinforcement Learning and Imitation Learning.

Reinforcement Learning

One of the possibilities is to train a model to play 1v1 Brawlhalla matches. By providing punishments and rewards I can make the model develop a certain policy to play 1v1 matches in Brawlhalla. The general idea is classic reinforcement learning, but since we can't reach the game memory and therefore can't reach the real-time game variables we have to try this another way. Instead of reading out live memory we can give the entire live gameplay to the model, and hopefully it will be able to learn how to play a 1v1 match of Brawlhalla



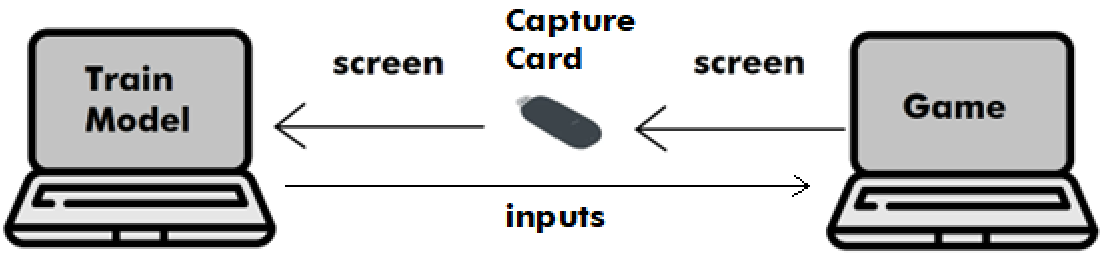
Environment

The environment will be the game itself. I will give the screen as input for the model and make it look at itself playing. This can be either done in 2 ways.

1. Make the model train on the same pc the game is being played.

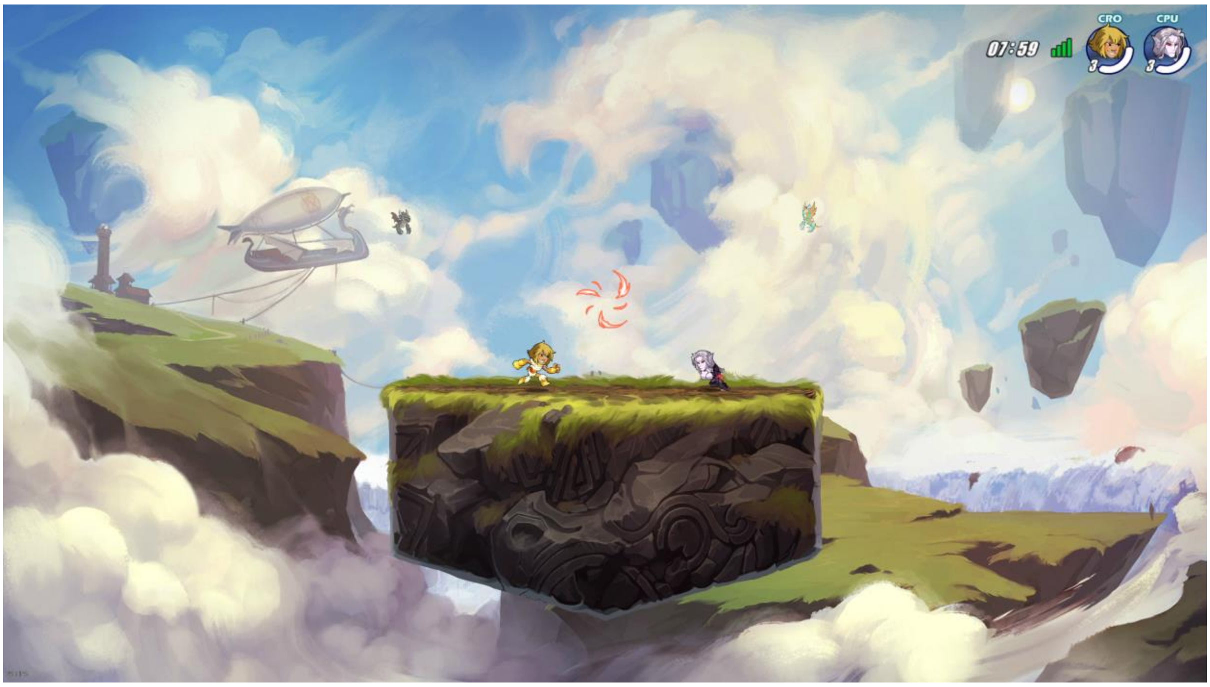
2. Make the pc with the model look at the screen of another pc which plays the Brawlhalla 1v1 matches using a Video Capture Card Device.

Important to note is that the Video Capture Card Device will add a small amount of delay to seeing the gameplay.

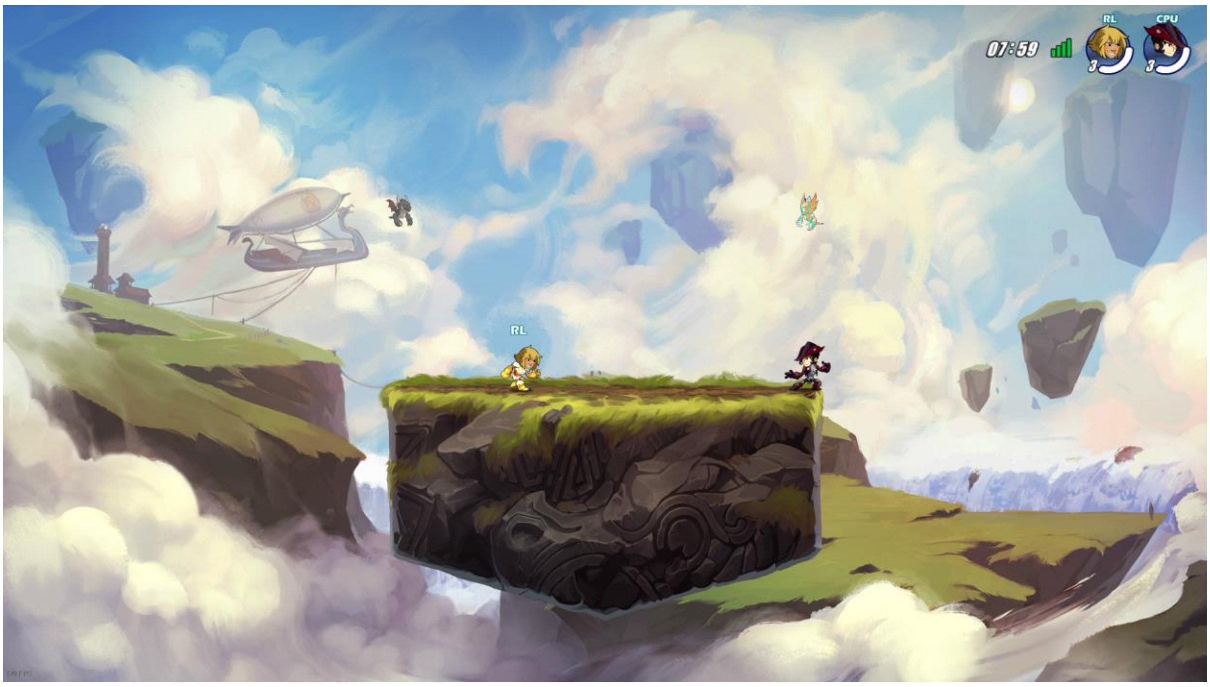


States

A state of the game is a frame of the footage I provide. I want the model to decide for each frame what to do. Here is an example of a game-state.



If we take this new state which is slightly different from the last state we inspected we can see that it now shows “RL” above our name. This “RL” will be displayed throughout the entire game each game until we turn it back off. This can be useful for the AI to make him understand who he is controlling. Like this we can try to make the game states as useful as possible.

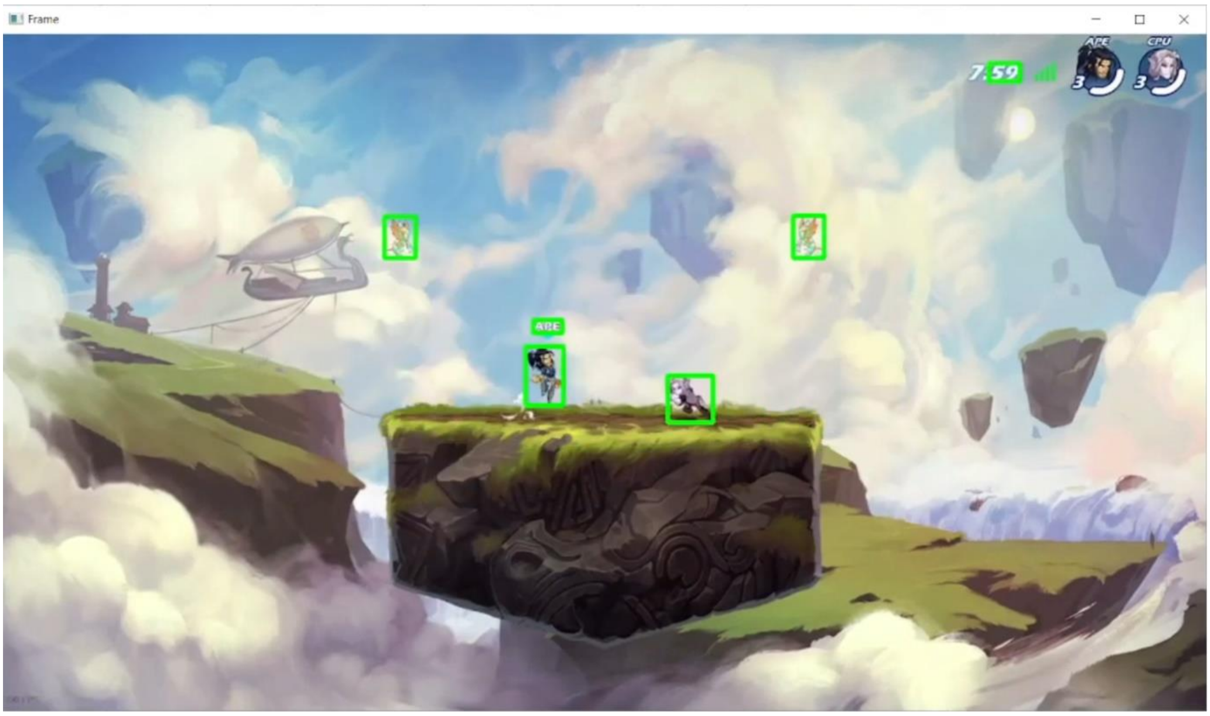


Clarify Game States with Object Tracking

It is also possible to try to extract game variables from the footage using object tracking. With object tracking we detect everything that is moving on screen and output 4 numbers which make up a bounding box, [x, y, width, height]. This means I can also give bounding boxes to the model to make game states more understandable.

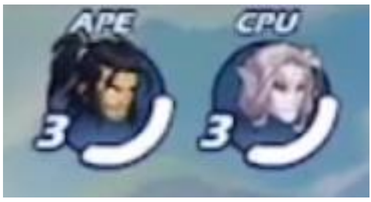
Which player is the Agent

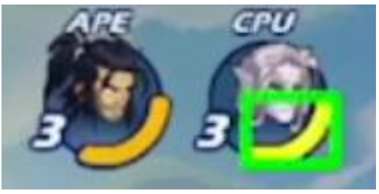
Here we have another game state. This time it shows the letters “APE” above our player as you can see. The result of this is that the bounding box of the player becomes bigger. This could make it easy for the ai to separate the 2 moving players/bounding boxes from each other. Not only that, but I can also recognize who is getting damaged, let me explain



Recognizing Damage

Here is a zoom-in on the health bars. Whenever someone is getting damaged the object detection will detect it and put a bounding box around it like the second image.

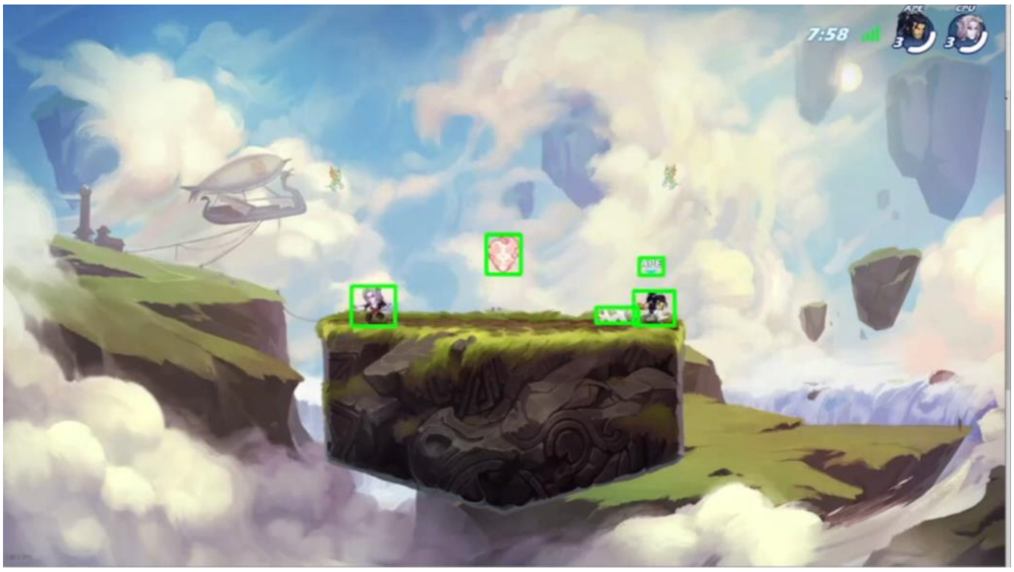


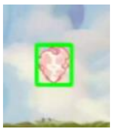


As you can see, there is a bounding box around the player with the letters “CPU” above it. These letters are different from our “APE” letters so the model could understand it is dealing damage right now, which is a good thing, so we can reward it.

Weapon Spawns

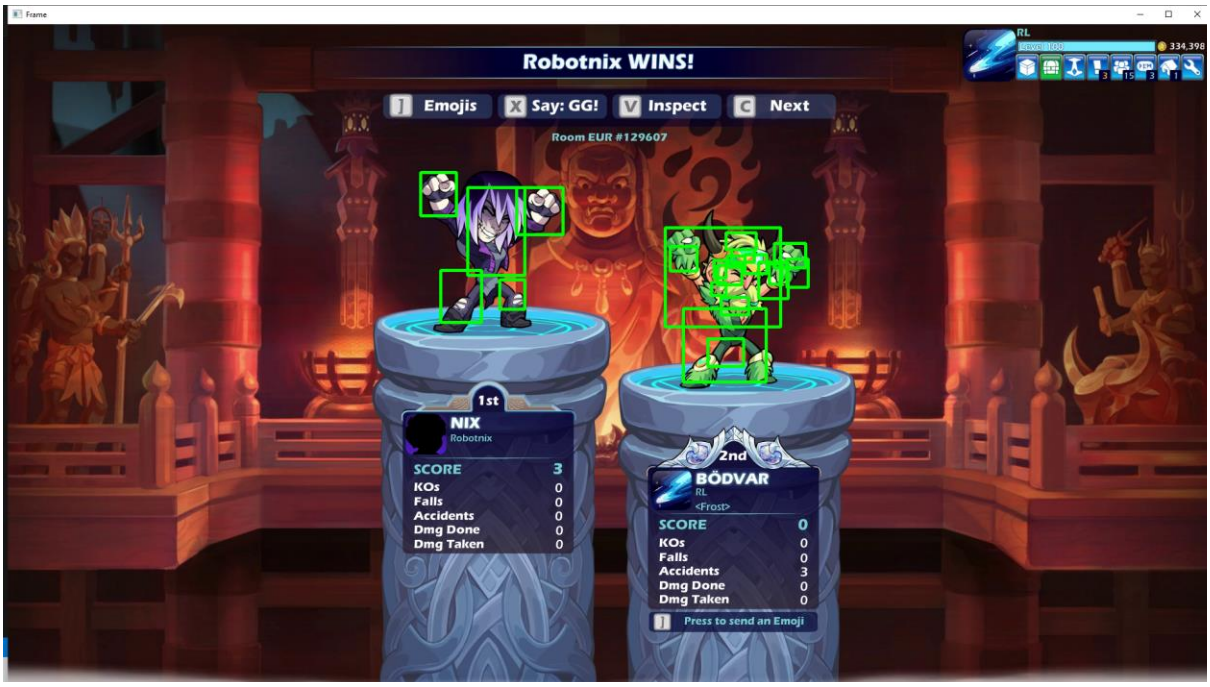
As last I need to mention weapons. Having a weapon in a macth of Brawlhalla 1v1 puts you in an advantegous stage since you have more range and more possible damage output. Luckily a weapon spawn is animated until it gets picked up or vanished. This means it will always have a pattern of bounding boxes around it which always will be the same. This means the model could recognize when a weapon spawns so it can pick it up and be rewarded for it



 The weapon Spawn.

Understanding Match Result

This one is very basic, in the current game state we see the match result shown after the match. If you are standing on the left and you are higher up, you won. If you are on the right and lower, you lost. Again it's ideal that it's animated until you advance so the bounding boxes will stay in that area.



SCORE	3
KOs	0
Falls	0
Accidents	0
Dmg Done	0
Dmg Taken	0

SCORE	0
KOs	0
Falls	0
Accidents	3
Dmg Done	0
Dmg Taken	0

Punishments and Rewards

To make the bot actually learn I need to provide some sort of punishment and reward.

Rewards

As rewards we can try to play around with the following events.

- Win a match

- Deal Damage

- Don’t take Damage

- Pick up a Weapon

Punishments

- Lose a Match

- Receive damage

All these event can be detected using Object Tracking which is ideal.

Actions

Here are all the actions the Agent can perform during a match. Keep in mind that each input can be given together with any other input. To hold an input the Agent simply used the input each consecutive frame.

W – Look up

A – Move Left

S – Move Down

D – Move Right

H – Pick up/Throw Weapon

J – Light Attack

K – Heavy Attack

L – Dash/Dodge

Spacebar - Jump

What is Next

I don’t know, this is where I am stuck right now. I don’t exactly know if just detecting the bounding boxes is enough, if I should give the values of the boxes, if I should feed anything to the Agent I don’t know.

Imitation Learning

Another possibility is to make a model learn from a Professional-Player. Imitation Learning is generally faster than reinforcement learning which is very useful in my case due to my limited time. Online there is plenty of gameplay of Professional-Players so data-wise it wouldn't be an issue either.

Environment

The environment will be the game itself. The Agent will try to replicate the Expert (Professional-Brawlhalla-Player) in a 1v1 match of Brawlhalla.

The setup of the environment has been mentioned earlier in this document, see Reinforcement Learning: Environment

States

See Reinforcement Learning: States

Punishments and Rewards

See Reinforcement Learning: Punishments and Rewards

Actions

See Reinforcement Learning: Actions

What is next

Again I don't know

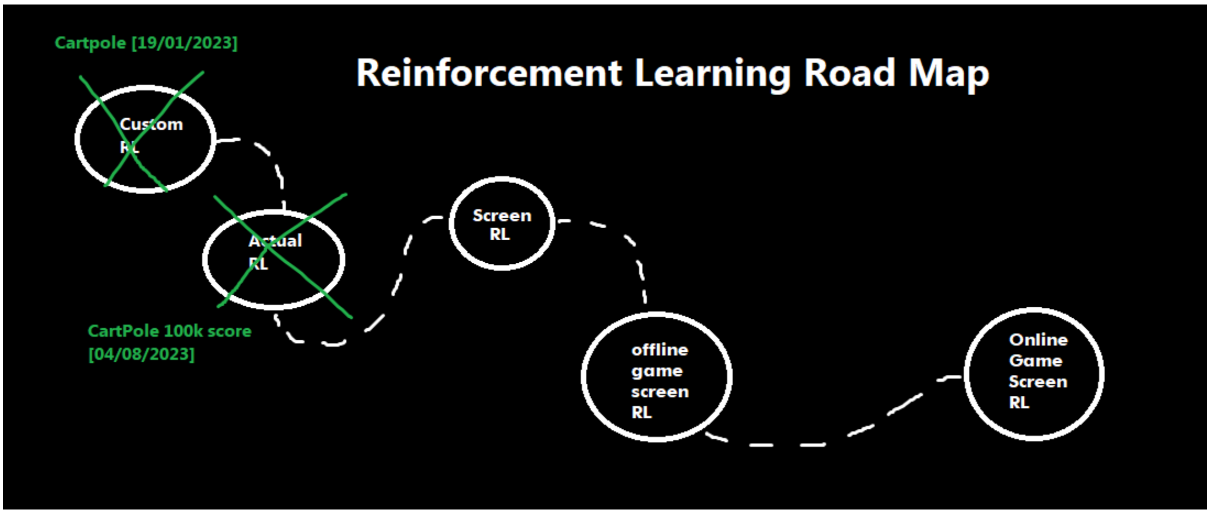
Comparison Table

	Reinforcement Learning	Imitation Learning
Speed	Can take very long	Quicker than RL
Effectivity	Can become worse than a Professional-Player but can also develop new strategies and become better than a Professional-Player	Become as good as an Professional-Player
Personal Foreknowledge	Oriënting to None	None
Data	Not Required	Plenty of data online available
Costs	Video Capture Card Device	None

So both seem worthwhile to try. I will try discuss all of this with my RL teacher to see which one is the more suitable one this semester.

Roadmap of Accomplishments

So to keep myself motivated and to give you a feeling of where I stand I want to make a little Roadmap. Here's the roadmap I made earlier as a hobby. Since it fits my current project perfectly I don't think I need to adjust it. I can shortly explain what the points in the roadmap mean.



Screen RL: Make a model that learns something by just watching the screen.

Offline Game Screen RL: Make a model that learns how to play an offline game by just watching the screen.

Online Game Screen RL: Make a model that learns how to play an online game by just watching the screen.

What are my current main Roadblocks?

1. I can replicate tutorials where they use reinforcement learning to train a model for a game or something else. But I can't find and figure out how I can give anything as input (image for example) and get my desired output. Make my own inputs, make my own outputs I can't find or do not know what to look for to learn it.

2. Teacher told me that I will probably need to make the states more understandable with image recognition for example how do you look into that.

3. Overall tips for the project?

4. Imitation Learning or RL?

References

Pysource. (2021, 28 januari). *Object tracking with OpenCV and Python* [Video]. YouTube. https://www.youtube.com/watch?v=O3b8lVF93jU

OpenCV: Getting started with videos. (z.d.). https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html

Tiger. (z.d.). *GitHub - Tiger767/ML-Brawlhalla: a machine learning bot that can 1v1 in the popular game Brawlhalla.* GitHub. https://github.com/Tiger767/ML-Brawlhalla

Timeolord. (z.d.). *GitHub - timeolord/Brawlhalla_RL_Agent*. GitHub. https://github.com/timeolord/Brawlhalla_RL_Agent

GeeksforGeeks. (2023). Multiple color detection in real time using Python OpenCV. *GeeksforGeeks*. https://www.geeksforgeeks.org/multiple-color-detection-in-real-time-using-python-opencv/

ZainBashir. (z.d.). *GitHub - ZainBashir/Solving-openAI-Gym-MountainCarProblem-using-DQN-with-Image-input: I have targeted to solve the benchmark problem in Reinforcement learning literature using Deep Q-networks with images as the only input to the model. Keras was used to model the Convolutional neural network which predicts the best action to take in a given state*. GitHub. https://github.com/ZainBashir/Solving-openAI-Gym-MountainCarProblem-using-DQN-with-Image-input