Dion James Watts-Evans [C1714994]
Supervisor : Federico Liberatore
CM3203 - One Semester Individual Project - 40 credits
Moderator : Charith Perera

# Initial Plan

## Machine Learning for Sonic the Hedgehog Video Games

# Project Description

## History and Context

Using machine learning to play video games has become something of a trend in recent years. Searching “AI learns to …” online yields countless results for almost any game you can think of. One of the pioneers for machine learning for gaming, is a now defunct flash game called Boxcar2d[1] back in 2011. The game used a genetic algorithm with roulette wheel selection the make cars that could go further and further along a progressively difficult track.

The machine learning for gaming craze really started to take off around 2015 with Sethbling’s MarI/O[2] a machine learning project using the Neuro Evolution of Augmenting Topologies or NEAT algorithm. Later they produced MariQ[3], a similar project that used Q learning to play Mario kart. These videos and the resulting craze are the direct inspiration for this project and have provide many resources to continue researching.

## Overview

The plan is to implement a Q learning algorithm for Sonic the Hedgehog games. Specifically, targeting the first level of the second game “Emerald Hill Zone act 1”. Focusing on a single level will hopefully result in a higher quality product. Human data will be gathered for this project as a point of comparison and other levels may prove difficult to acquire good information.

To implement the model, the plan is to use the game screen as an input and use reinforcement learning for a given game state to develop the model. This is going to be done by extracting game data from the emulator in real time. Making a model that can beat the level should not be too much of an issue as Sonic games are designed to be inherently easy to beat but the difficulty lies in beating them quickly. As such the main aim is to produce a fast AI.

## Technical

Sonic 2 will be running in the Bizhawk Emulator[4] developed by the Tool Assisted Speedrunning(TAS) community for producing Fast playthroughs of games using tools to demonstrate what is hypothetically possible. These tools are useful for my use case as they allow me to extract game data and input controls with precise timing. Most importantly Bizhawk contains a Lua scripting interface which I will use to interface with my Python Q learning implementation.

Q learning was picked as the algorithm of choice as it seems relatively underutilised in the scene. Most examples found use either NEAT or a similar Recursive Neural network. No specific implementation of Q learning for Sonic was found, However this paper by openAi[5] about using Reinforcement learning to clear sections of various sonic games gives me some hope in the use of Reinforcement learning.

Q learning in Python was chosen due to its wide variety of machine learning libraries and general familiarity. Research on these libraries has begun and if all goes to plan, TensorFlow will be used to implement the project.

# Project Aims

## Aim 1 – Level Completion

After implementing the basic machine learning framework, the first aim is to create a model that is capable of reliably beating the level. Reaching this aim means that the basic framework of the game state model and the rewards system for reinforcement learning has been implemented sufficiently.

**Research Question:** Can Q learning be implemented to play Sonic the Hedgehog games?

## Aim 2 – Speed

Sonic games pride themselves on the speed that comes with replayability. The ultimate goal is to beat the level as fast as possible. Thus, after the level is beat, the goal becomes optimising the Ai to overcome fit into higher categories of speed:

1. Human Players that are not familiar with gaming at all
2. Human Players that are familiar with gaming but not with Sonic
3. Human Players that are familiar with Sonic
4. Human Players that speedrun Sonic
5. The optimised tool assisted speedrun or hypothetical best

These categories get harder to obtain as it progresses, and later categories are unlikely to be obtained.

most of this data will be obtained via live data gathering from human subjects, however some of the data will obtained from open data online.

**Research Question:** How does the Q learning Model compare with real players?

## Optional Aim – Generalization

One week of the work plan is allotted to experimentation. This is to test the model created on "Emerald Hill Zone" on other levels and games. It will be interesting to see if any level of generalization can be achieved and should provide for possible discussion.

**Research Question:** Can generality be achieved from one level of training data?

# Ethics

There were originally some concerns over the ethics of using Emulation for this project considering their dubious nature. However, there are many legal precedents that emulators fall under fair use, such as the case of Bleem![6]. More importantly the Rom for Sonic the Hedgehog 2 was legally acquired via it's recent PC release.

This project will conduct an experiment with volunteered participants who will play the level the model will be training on to compare their gameplay to the model's. the participants will be split into groups based of off experience and will play through the level a few times noting their best and worst time. Video footage of some of these playthroughs will be recorded, though no identifying information should come through gameplay capture.

As such the project has been planned understanding the time needed to consult the ethics board about this experiment.

## Work Plan

The work plan is split over 15 weeks, I have allotted buffer time to the last week and easter break. I have split the plan into Sections, Goals and Tasks. Sections describe the project in vague terms, Goals should be achieved by the end of a given week, and Tasks are a measure of Goal completeness.

I will catch up with my supervisor biweekly starting on week 2. Due to scheduling issues, the week highlighted in green represents the Goal that should be completed before a supervisor review.

| Week | Section | Goal | Tasks |
|---|---|---|---|
| 1 | Research and planning | • Complete and submit the initial plan. | • Research similar projects<br>• Decide what Machine learning algorithm to use.<br>• Begin developing understanding of necessary ML concepts.<br>• Write and Submit Initial Plan. |
| 2 | Python – Bizhawk interfacing | • Understand Q learning concepts.<br>• Develop a system to obtain information about the game state using Lua. | • Find sources of information on Q learning and further develop understanding.<br>• Discover which bytes of RAM/game code is useful for representing the game state.<br>• Create a tool using Lua that passes this information into Python. |
| 3 | | • Create a Representation of the game state in Python that updates in real time. | • Using the Lua interface obtain information about the game state and use that to interpret information about the player's position.<br>• Allow python to input controls into Bizhawk. |
| 4 | Implementing Q learning for Sonic the Hedgehog in Python | • Begin implementing Q learning in python. | • Create Reward system for a given game state for reinforcement learning.<br>• Implement Q learning with some Python ML library (likely tensorflow). |
| 5 | | • Have a presentable basic | • Finalise basic implementation.<br>• Create some work in progress deliverables to demonstrate progress. |

| | | | |
|---|---|---|---|
| | | implementation of the project | |
| 6 | | • Tweak implementation following supervisor feedback | • Conduct review meeting with supervisor.<br>• Begin implementing changes discussed in review meeting. |
| 7 | | • Finalise implementation ready to produce deliverables | • Tweak changes to optimise model.<br>• Decide on the trained model I am going to use for demonstration purposes. |
| 8 | Producing deliverables and experimenting | • Creation of deliverables for the report (i.e., Charts) and supporting resources (i.e., videos)<br>• Experiments with the generality of the created model | • Turn collected data into a usable form.<br>• Create figures for use in the report.<br>• Export example videos.<br>• Experiment with the model in different cases, such as a different level or Sonic game. |
| E1 | | | |
| E2 | | | |
| E3 | | | |
| 9 | Producing the report | • Outline and begin writing the report | • Gather resources necessary for the report, collect references, finalise figures etc.<br>• Outline the report, making sure all major sections exist.<br>• Begin writing. |
| 10 | | • Report is nearing completion | • Continue writing the report.<br>• Assure report is in a presentable state to supervisor for review meeting. |
| 11 | | • Finalise report | • Conduct review meeting with Supervisor.<br>• Begin implementing changes discussed in review meeting.<br>• Review and Submit. |
| 12 | | • Report buffer week | • Exists as buffer time if necessary. |

# Bibliography

[1] (25 January 2011). Boxcar2d. available at: http://boxcar2d.com/ (Accessed: 4 February 2021)

[2] Sethbling(Jun 13, 2015). MarI/O - Machine Learning for Video Games. Available at: https://youtu.be/qv6UVOQ0F44 (Accessed: 05 February 2021)

[3] Sethbling(Jun 29, 2019). MarIQ -- Q-Learning Neural Network for Mario Kart. Available at: https://youtu.be/Tnu4O_xEmVk (Accessed: 05 February 2021)

[4] (2021-01-17) Bizhawk. available at: http://tasvideos.org/BizHawk.html (Accessed: 5 February 2021)

[5] Nichol A., Pfau V., Hesse C., Klimov O., Schulman J. (2018) Gotta Learn Fast: A New Benchmark for Generalization in RL. available at: https://arxiv.org/pdf/1804.03720.pdf (Accessed: 5 February 2021)

[6] Sony Computer Entertainment Inc. v. Connectix Corp.(2000) 203 F.3d 596