NeuronLeap

Degree programme: BSc in Computer Science | Specialisation: Computer Perception and Virtual Reality
Thesis advisor: Prof. Dr. Jürgen Eckerle
Expert: Dr. Federico Flueckiger (Eidgenössisches Finanzdepartement EFD)

This April OpenAi Five beat the world-champions in a best of three Dota 2 match. To achieve this OpenAi used a team of five neural networks they trained with reinforcement learning by playing against themselves.

Using a method similar to theirs the goal was to recreate their success in a self-made multiplayer jump‘n‘run for Android.

Context

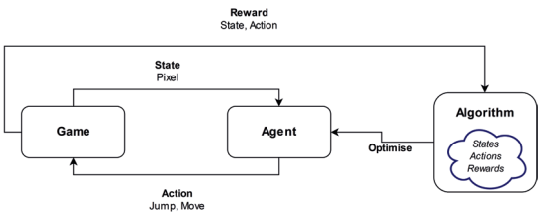
In recent years deep reinforcement learning has made huge advances. Alpha Go beat the best player in Go only three years ago and now in 2019 already the world champions in Dota2 were beat, a game that much more complex. Such news isn't only exciting for artificial intelligence in games, the same methods can be used for real world problems and application like controlling a robotic hand or planning energy efficient and fast traffic routes for a fleet of trucks.

Goal

The aim of this thesis is to extend the single player prototype that was created during a prior project to a multiplayer version. It was stipulated that the game should be playable by running it on two phones with a tablet functioning as server. The target then is to provide an AI that can substitute the second player. Said AI is to be trained with a reinforcement learning method.

Game

NeuronLeap is a multiplayer jump'n'run for Android where the goal is to roll and hop a ball to the goal faster than your opponent, without falling down or being eaten by the left side of the screen.
It was created with the help of libGDX, a cross-platform game development framework and lots of other tools to make maps, buttons and physics.
To make use of the multiplayer modus the players just have to be in the same network as a server running the application.



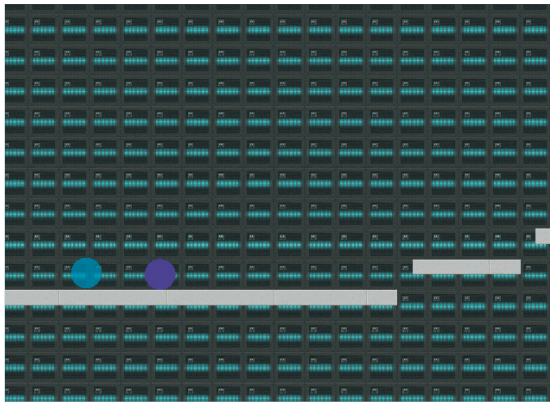
A very simplified version of the training process

Network

The Neural Network was trained by playing against itself. It learned with the help of a method called deep Q-Learning. In this method the algorithm gets the pixels and the possible actions as an input as well as a score after each action. It then tries to maximise the score by optimizing the neural network to take the best action depending on the state. To figure out what the best action is the last few states, actions and rewards are retained. This “short term memory” is useful because it usually takes a series of actions to achieve a reward.

Result

The result is a working two player jump'n'run that can be played against an AI. While the AI did learn some interesting strategies like sabotaging the other player by pushing him off it remined somewhat unintelligent overall and didn't achieve a human level of control.



Snapshot of the training

Alexandra Dominique De Groof