Artificial Intelligence Systems in Video Games

By Jayson Demarchi, Thomas Lai, Thomas Perrodin, Gregory Petrillo, Jude Woo

TEAM 3

Citation

- Real-time Heuristic Search for Pathfinding in Video Games by V.Bulitko, Y.Bjornsson, N.R. Sturtevant, R.Lawrence
  http://www.ru.is/~yngvi/pdf/BulitkoBSL10.pdf

- https://en.wikipedia.org/wiki/Artificial_intelligence_(video_games)

- Introducing: Unity Machine Learning Agents Arthur Juliani, September 19, 2017
  https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/

- Artificial Intelligence for Games: Decision Making by Dave Mount
  https://www.cs.umd.edu/class/spring2013/cmsc425/Lects/lect20.pdf

- Black and White Case Study by James Wexler
  https://www.cs.rochester.edu/~brown/242/assts/termprojs/games.pdf

- MarI/O - Machine Learning for Video Games
  https://www.youtube.com/watch?v=qv6UVOQ0F44

- Entertainment Computing and Serious Games: International GI-Dagstuhl Seminar 15283, Dagstuhl Castle, Germany, July 5-10, 2015, Revised Selected Papers
  http://pcg.wikidot.com/

What we will cover

1. A introduction to A.I. application in the video game domain
2. Some uses of Video Game A.I.
3. Adaptive A.I. system in video game
4. Machine learning agents helping to train A.I.

   in video games
5. Marl/O Case Study

History

- 1970s: Pong, Space Invaders popularized computer-controlled opponents
  - Space Invaders used random movement, no decision-making
  - Pong at least needed to track the ball
- 1980: Pac-Man released
  - Featured simple pathfinding and varied ghost behavior
- 1990s: RTS (Real Time Strategy) games emerge
  - Increased complexity involving incomplete data, continuous decision-making, pathfinding, etc.
- 2000s: Game AI sophistication explodes
  - Black and White (2001): first use of reinforcement learning in video games
  - More games featuring more intelligent agents
  - Coincides with huge industry growth

History

- 2010s: AI got good at playing games
  - Google’s DeepMind and Elon Musk’s OpenAI release free tools for AI development and training
  - Researchers drawn to video games as great sandboxes for AI training and testing
  - Game developers like Blizzard begin releasing apis to support AI research with their games

https://techcrunch.com/2017/08/09/blizzard-and-deepmind-turn-starcraft-ii-into-an-ai-research-lab/

Video Game AI vs Game playing AI

- Video Game AI
  - NPCs
    - Non-player characters
- Game Playing AI
  - MarI/O
    - Mario Bros AI utilizing genetic algorithms
  - OpenAI Dota 2 AI
    - AI that recently defeated professionals in Dota 2

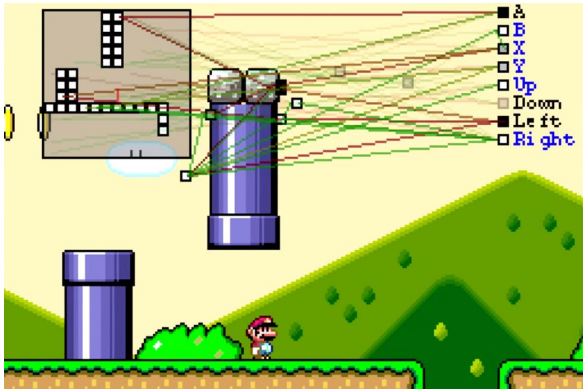
https://youtu.be/rWFPw8Lt1bk

OpenAI

https://openai.com/

Game Playing AI

- Generally developed after the game
- Stand-in for human player
- Information available = information available to player
- Usually a separate process from the game
  - Sometimes even running on separate hardware
- The challenge is making them play well



https://www.engadget.com/2015/06/17/super-mario-world-self-learning-ai/

http://vizdoom.cs.put.edu.pl/competition-cig-2017

Video Game AI

- Developed as part of the game
- Control “bots” and “NPCs”
- Common in “PvE” gameplay
  - Player vs environment
- Can cheat
  - Unlimited knowledge of game state
  - Potentially perfect accuracy
- Classifier is part of the game
  - Classification is part of the game process
- Challenges include
  - making them move/act like a human/intelligent being
  - strike difficulty balance
  - (appear to) play by the rules
  - Not too computationally intensive

https://i1.wp.com/enterinitials.com/wp-content/uploads/2017/01/bokoblin.gif

Artificial Intelligence Systems in Path Making for RTS games

- RTS games = Real Time Strategy games
- Popular examples: Starcraft, Warcraft III, Age of Empires... etc
- One major feature of RTS is the path finding
- Select a unit, and click on a location
- To reach that location, the unit will have to follow a path around obstacles and it should the most optimal path

A* pathing

- To calculate the optimal route from point A to B in real time, the bot use A* heuristic search

- Calculates several possible moves, one move at a time
  Each move has a cost, and to find the optimal, the cost should be minimized

- Attributes to check - Collidable? Distance from destination?

Learning Real-Time A*

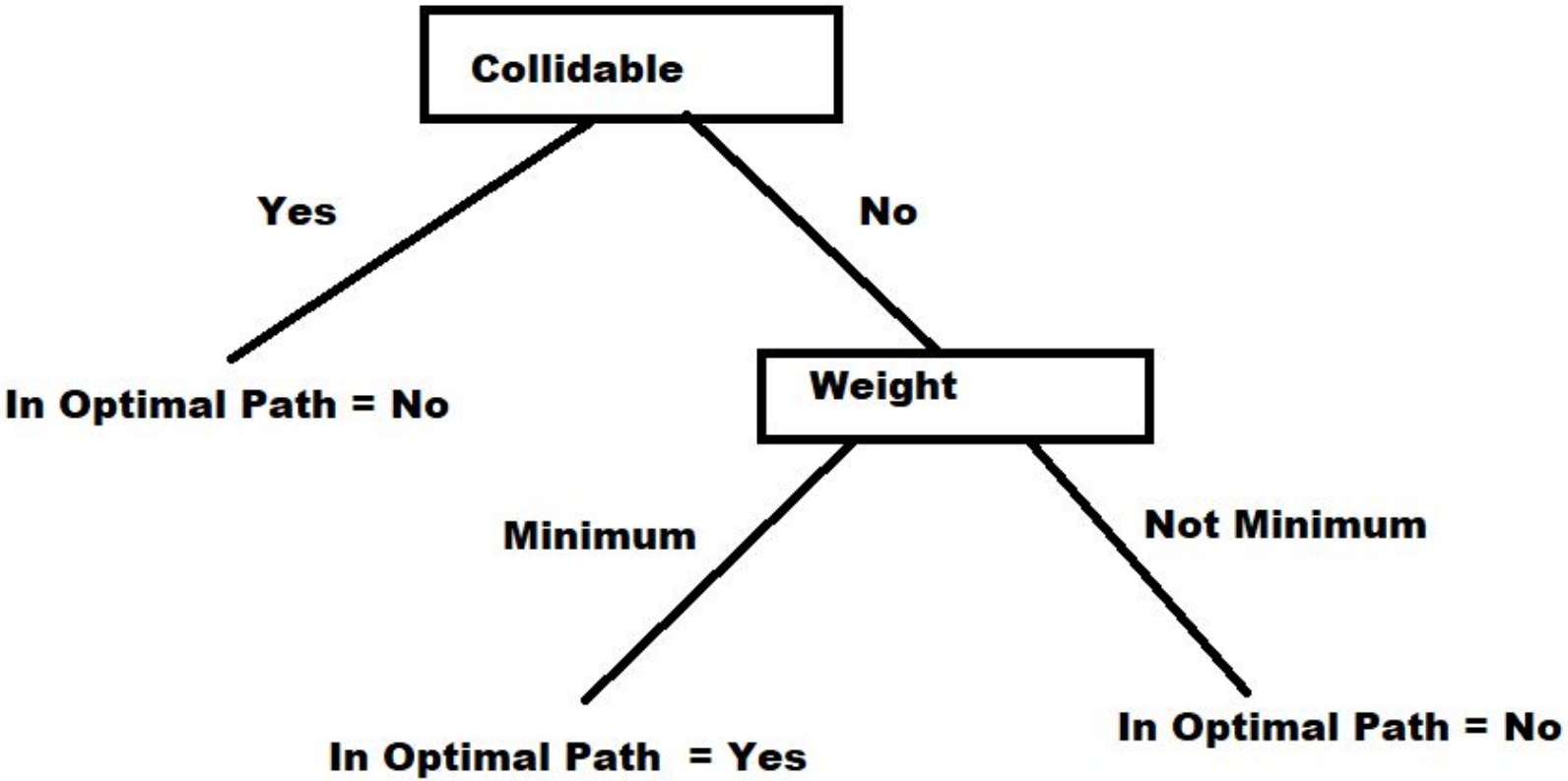
LRTA*$(s_{\text{start}}, s_{\text{global goal}}, g_{\max})$

1 $s \leftarrow s_{\text{start}}$
2 **while** $s \neq s_{\text{global goal}}$ **do**
3 select a subgoal s_{goal}
4 generate successor states of s up to $g_{\max}$ cost, generating a frontier
5 find a frontier state s' with the lowest $g(s,s') + h(s', s_{\text{goal}})$
6 update $h(s, s_{\text{goal}})$ to $g(s,s') + h(s', s_{\text{goal}})$
7 change s one step towards s'
8 **end while**

Figure 2: LRTA* algorithm extended with dynamic subgoal selection.

Source; Real-time Heuristic Search for Pathfinding in Video Games by V.Bulitko, Y.Bjornsson, N.R. Sturtevant, R.Lawrence

A* tree - at each subgoal



Video Game AI decision making

Wandering - uses a rule based system

Ahead	Right	Left	Action
Open	–	–	Go ahead
Blocked	Open	–	Turn right
Blocked	Blocked	Open	Turn left
Blocked	Blocked	Blocked	Turn around

Fig. 1: A (ridiculously) simple wandering behavior for a ghost in Pac-Man.

Source: https://www.cs.umd.edu/class/spring2013/cmsc425/Lects/lect20.pdf

Video Game AI decision making

Finite State Machines

- One state, check if what is true

- Transition to different state

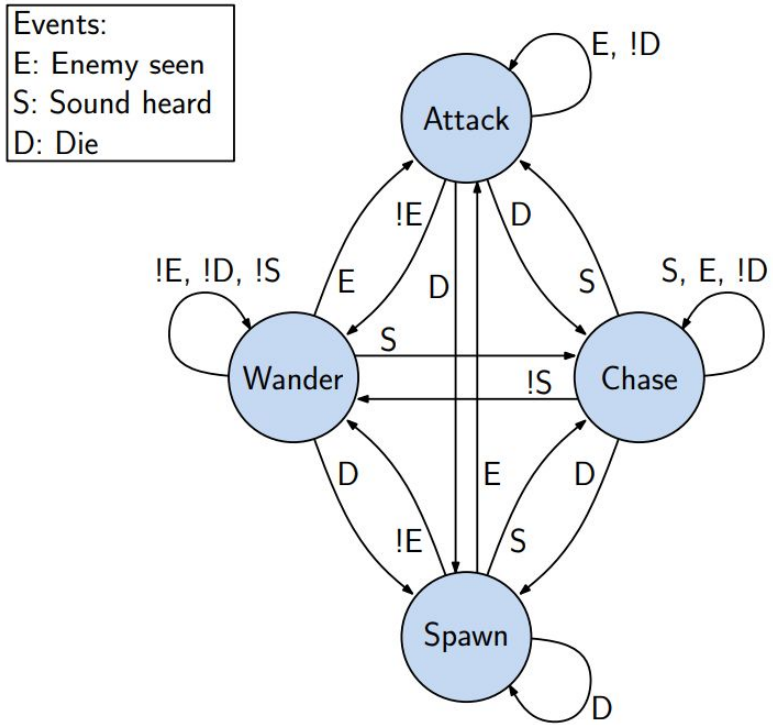


Fig. 2: Implementing an enemy combatant NPC for a FPS game.

Source:https://www.cs.umd.edu/class/spring2013/cmsc425/Lects/lect20.pdf

Improvements to be made

- The A.I. system does not take everything into consideration, or does not have rules for some things

- Increasing a game's difficulty could possibly mean

- Dialing up an A.I.’s learning efficiency and ability Scales the challenge factor up smoothly

Source: https://i.imgur.com/wZ1ql47.jpg

Background of adaptive A.I. system

- Adaptability in games - artificially intelligent NPCs can be central to gameplay design and is important for the player experience.

- Modus operandi - adaptive A.I. is needed to be built around creating a more engaging game for the individual player.

- Importance of player psychology to adapt to - potentially provide improved control of the game system & more tailored experience for individual players.

https://arxiv.org/pdf/1607.05028.pdf

Principles of adaptive A.I. system

Adaptive A.I. involves leveraging information from the ’User(s)’ to dynamically alter the elements of the central ’System’ module.

Thus, regulating activity within the right-hand ’Experience’ module, which feeds back to the ’User(s)’.

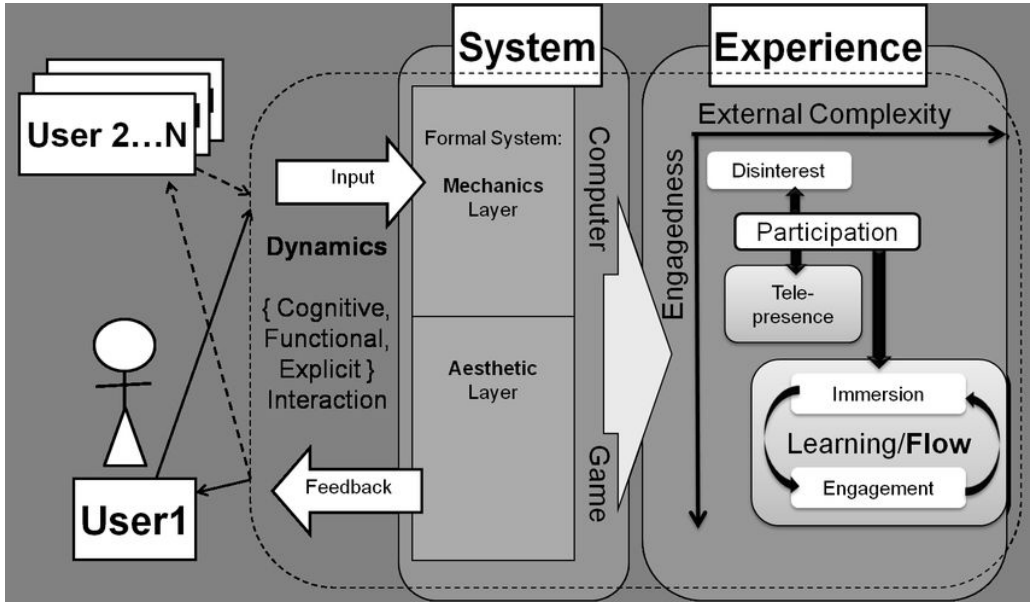


Fig. A model of player interaction with a computer game, for two or more players
https://www.researchgate.net/figure/305401165_fig1_Fig-1-A-model-of-player-interaction-with-a-computer-game-for-two-or-more-players

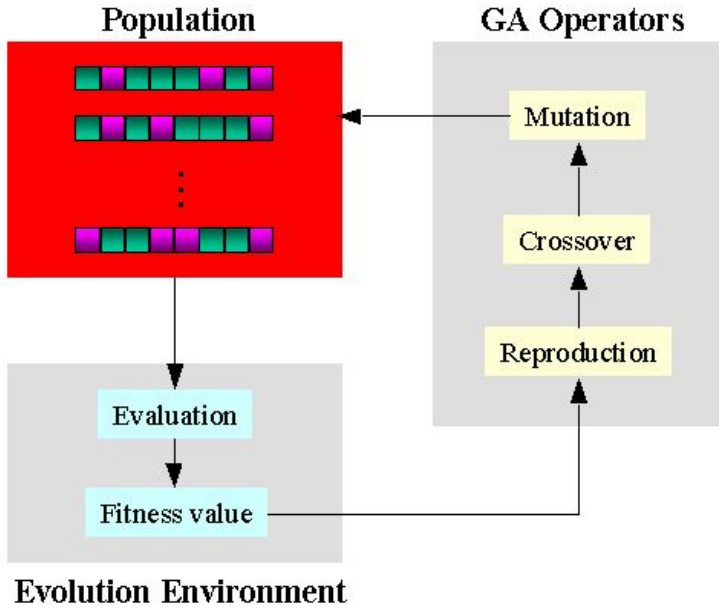
Techniques of implementing adaptive A.I. system

- **Dynamic Difficulty Adjustment (DDA)** - automatically changes parameters based on the player's ability.
- **Player Experience Modelling (PEM)** - adopts the game environment to the player’s capabilities and preferences.
- **Procedural Content Generation (PCM)** - generates programmatic of game content that results in an unpredictable range of possible game play spaces.
- **Massive-Scale Game Data Mining (MDM)** - extract and transform massive information from a game into an understandable structure.

Adaptive A.I. system using genetic algorithm

- **Genetic Algorithm** - A population-based search techniques that maintain populations of potential solutions during searches.
- **Adaptive A.I.** can be implemented by **GA**, following these steps:

  Define the possible solutions
  Create an **initial** generation
  Calculate relative fitness values
  Create a **new** generation
  Calculate relative fitness values (**AGAIN!**)



Genetic Algorithm Evolution Flow

http://www.ewh.ieee.org/soc/es/May2001/14/Begin.htm
https://www.gamasutra.com/blogs/MichaelMartin/20110830/90109/Using_a_Genetic_Algorithm_to_Create_Adaptive_Enemy_AI.php

Application using adaptive A.I. system 1

https://gfycat.com/gifs/tag/GTA-V

Application using adaptive A.I. system 2

Hosted on the Twitch channel Sentdex, “Charles” is an **adaptive artificial intelligence** that learns by the trial-and-error approach.

Charles is trying to learn how to “**drive**” in this game with various attempts.

https://kotaku.com/i-cant-stop-watching-a-gta-v-ai-learning-how-to-drive-o-1795371975

Future of adaptive A.I. system

A two-pronged approach for game designers to effectively implement adaptive game AI:

- Adapt a game at the mechanics layer
- Give the formal specifications of in-game adaptive AI
- Make players want to play a game more as they play

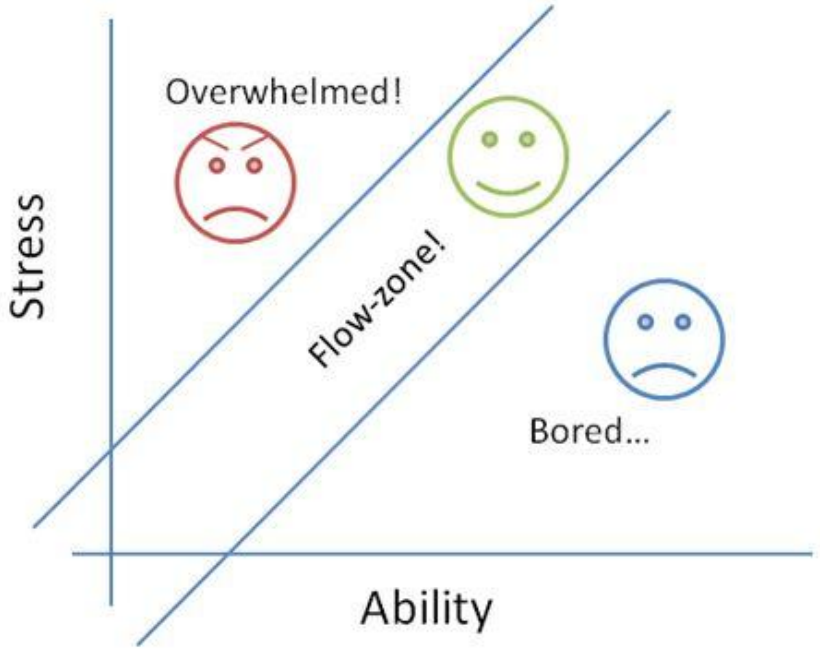
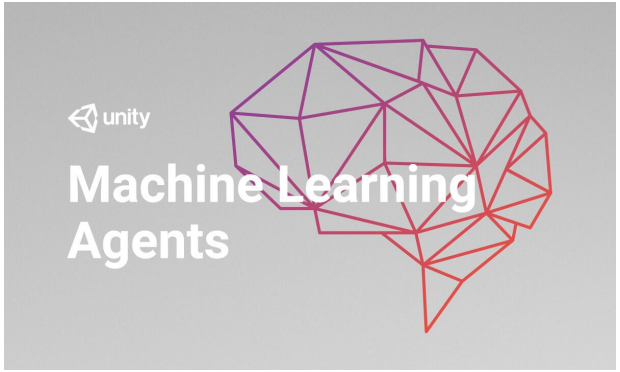


Fig. A player's mood on one's in-game ability & the amount of stress from playing
https://www.gamasutra.com/blogs/MichaelMartin/20110830/90109/Using_a_Genetic_Algorithm_to_Create_Adaptive_Enemy_AI.php

Machine Learning of Video Game A.I.

Machine Learning is constantly improving the intelligent behavior of video game A.I.

Unity has developed their own machine learning agents to allow researchers and developers to create games and simulations where A.I. can be trained with reinforcement training.



Source:https://blogs.unity3d.com/wp-content/uploads/2017/09/ML-blog-header-v6.jpg

unity

Unity ML - Agents (Beta)

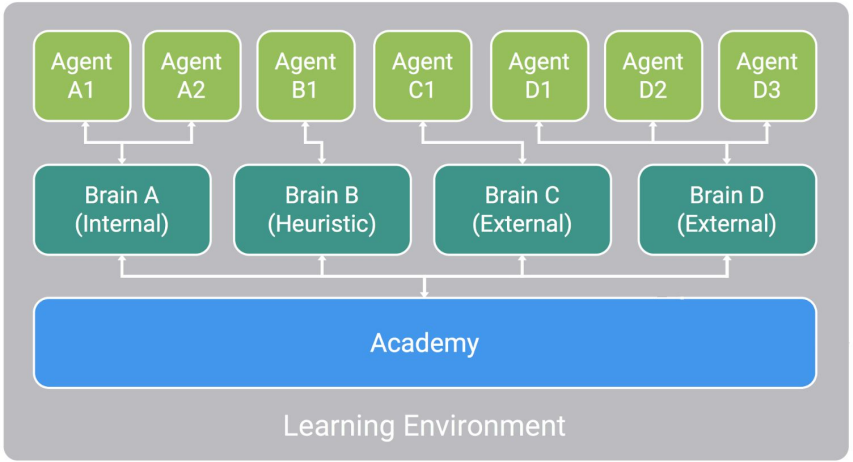
Source:*https://github.com/Unity-Technologies/ml-agents*

Three Main Parts to a Training Environment

The **Agents** - characters that acts on the environment

The **Brains** - controls the actions of the agents

The **Academy** - Only one per environment, it controls all settings in the environment. (speed/render quality, etc.)

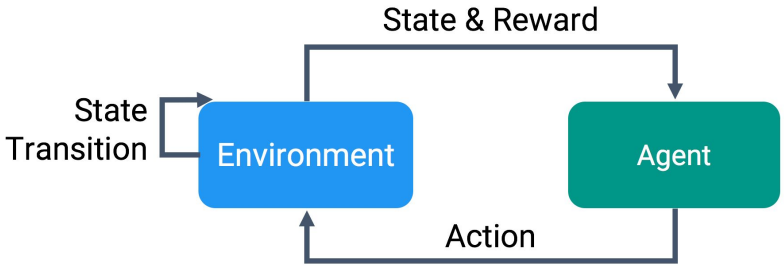


Source:https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/

Teaching A.I. Through Training Environments

The A.I. agent in the game development tool is tied closely to a learning environment, where every action results in a change of state on it.

When a change in the state of the environment is one defined as desired, the Agent is ‘rewarded’, and is likely to try that action again when trying to achieve its goal.



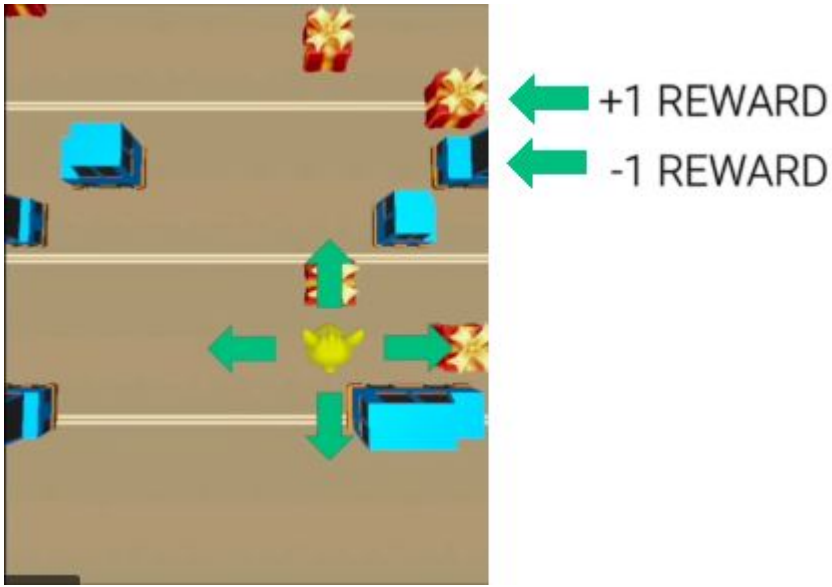
The typical Reinforcement Learning training cycle.

Source:https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/

Training a Single Agent

This single agent is able to move in 4 directions. Its goal is to move through the rows of blue trucks without getting hit, and to collect presents along the way.

The presents provide +1 reward, and the trucks provide -1 reward.

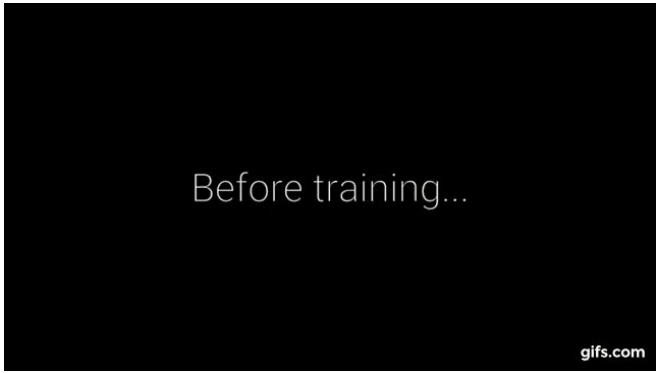


Source:https://www.youtube.com/watch?v=fiQsmdwEGT8

Results of the training

Before training, the agent was able to get one or two presents at the beginning where the trucks were slowest.

It showed steady progress and managed to get more than 110 presents after only 6 hours of training.

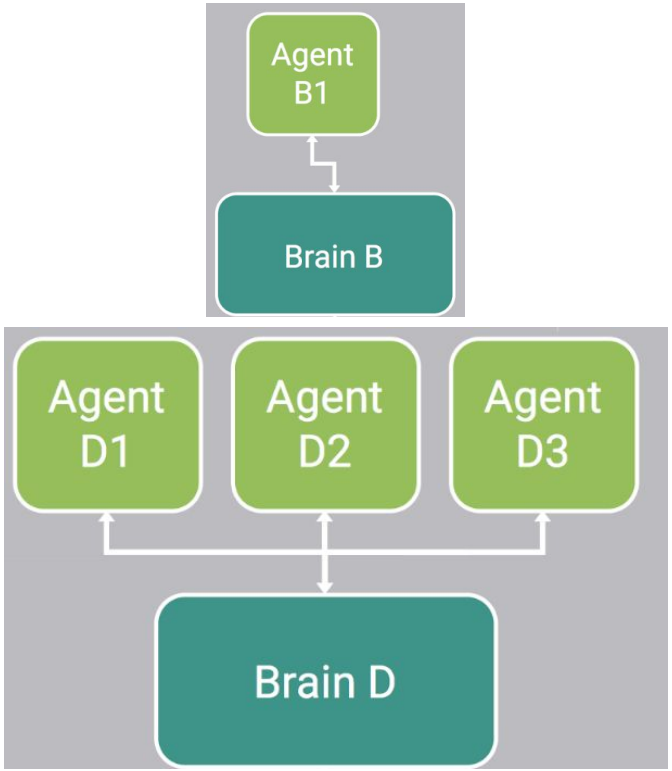


Source:https://www.youtube.com/watch?v=fiQsmdwEGT8

Training Using Multiple Agents At Once

In the previous slides, the Agent made great progress in a few hours, but every improvement it had to make on its own.

Next, we'll see an example of **Simultaneous Single-Agent**, where 12 agents are all linked to a single brain, and learn from each other.

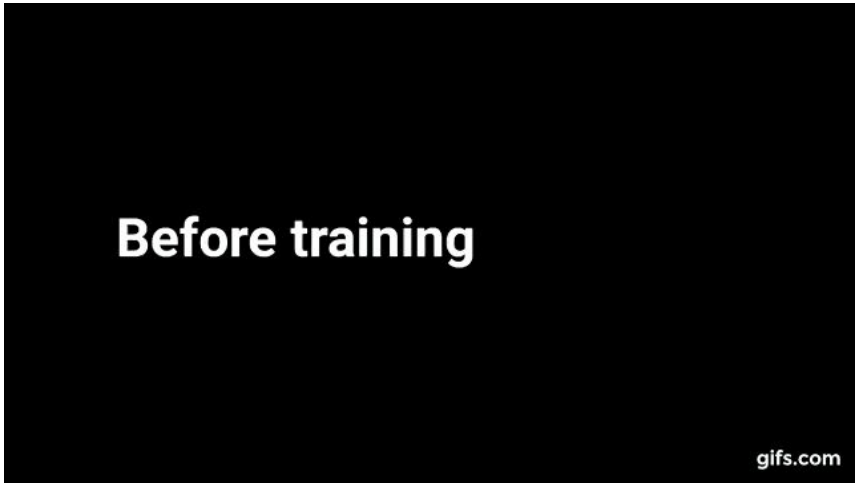


Source:https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/

What Drives These Agents?

Each of the agents has a goal: Learn to balance a ball on a platform using Reinforcement Learning.

By parallelizing the agents, they were able to balance a ball consistently in only 30 seconds.



Source:https://www.youtube.com/watch?v=fq0JBaiCYNA

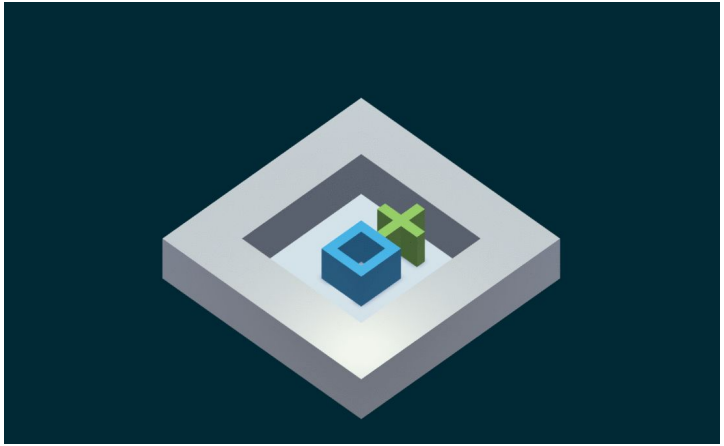
Brief Look At Other Available Training

- **Adversarial Self-Play** - Two agents with inverse rewards linked to a single brain. This allows for training of A.I. in two-player games. This is the same strategy used to train the AlphaGo A.I. and the more recent OpenAI to train a DotA 2 bot.
- **Cooperative/Competitive Multi-Agent** - Many agents with a shared reward, where all agents work to complete a task that can't be completed alone, sometimes against another group of agents.
- **Ecosystem** - Multiple agents with independent rewards that are put into the same environment.

Additional Features

Curriculum Learning - Sometimes, a task may be too difficult for agents to learn outright. By putting the Agent on a curriculum, you can start it at an easier difficulty, and gradually increase difficulties as it progresses.

Doing this allows for more efficient learning. In ML-Agents, custom environment parameters can be used any time the environment resets for another run. This means that difficulty/complexity elements can be dynamically adjusted.

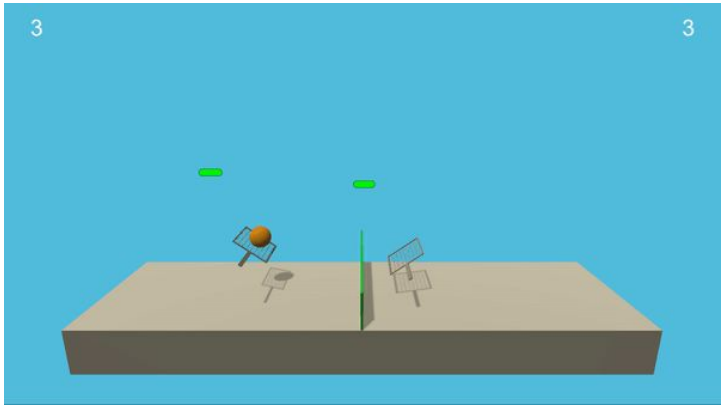


Source:https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/

Monitoring Agent Decision Making

Since communication in ML-Agents is a two-way street, they provide an Agent Monitor class in Unity which can display aspects of the trained agent, such as policy and value output within the Unity environment itself.

Here each agent shows how much future reward the agent expects. When the right agent misses the ball, the value estimate drops to zero, since it expects the game to end soon, resulting in no additional reward.

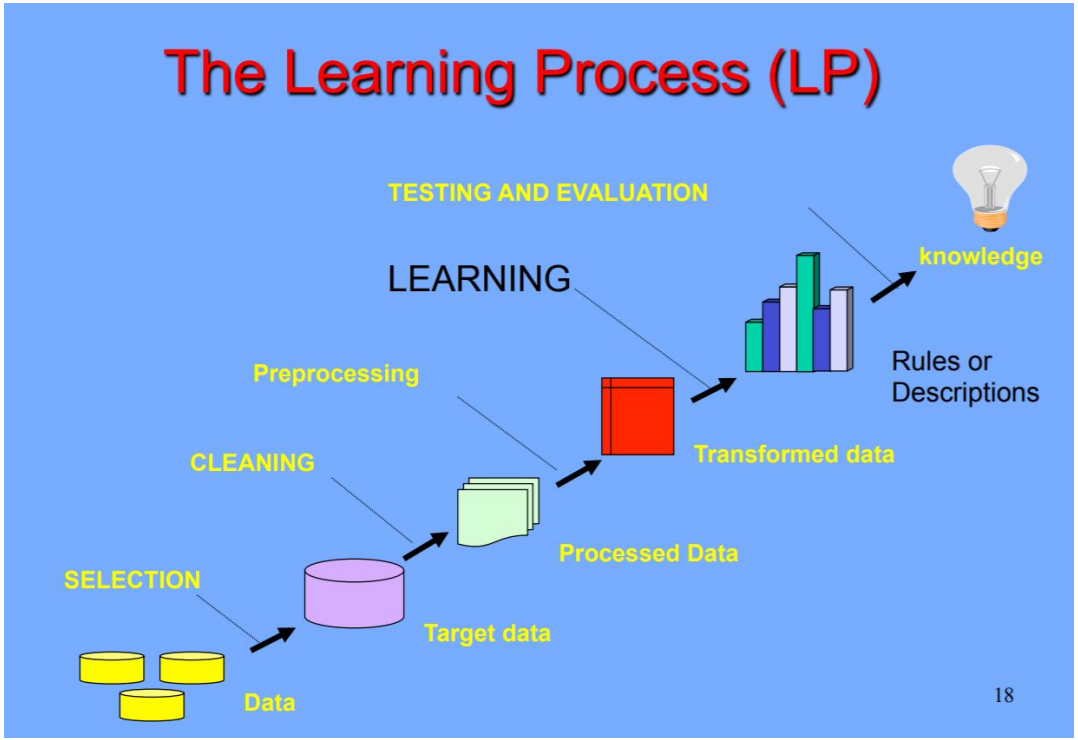


Source:https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/

MarI/O by SethBling

Source: http://bit.ly/2y6WpGb

Selection of Data

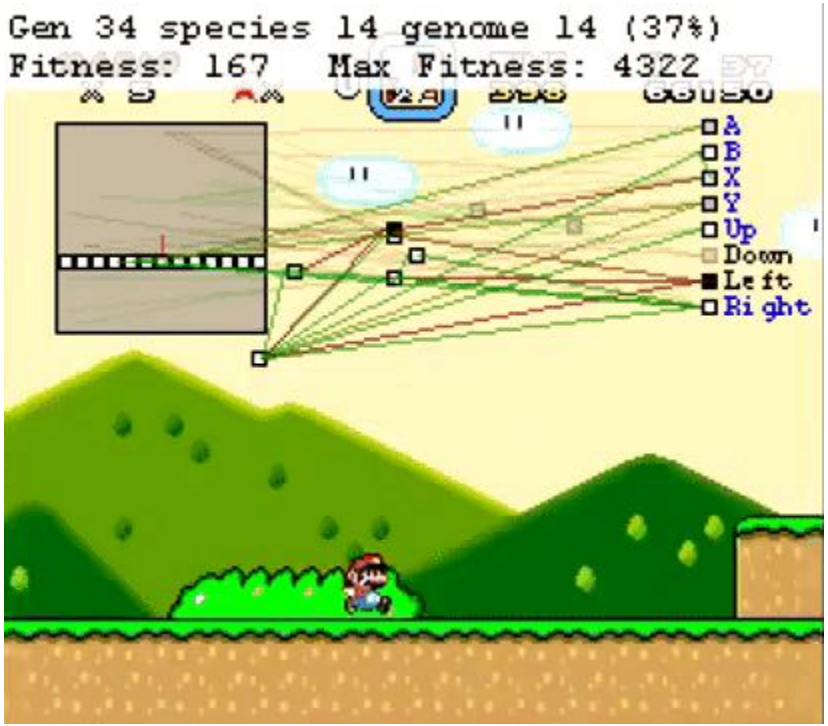


Source: http://www3.cs.stonybrook.edu/~cse352/L12NN.pdf

Preprocessing of Data

White blocks are blocks Mario can stand on

Black blocks are moving objects or enemies



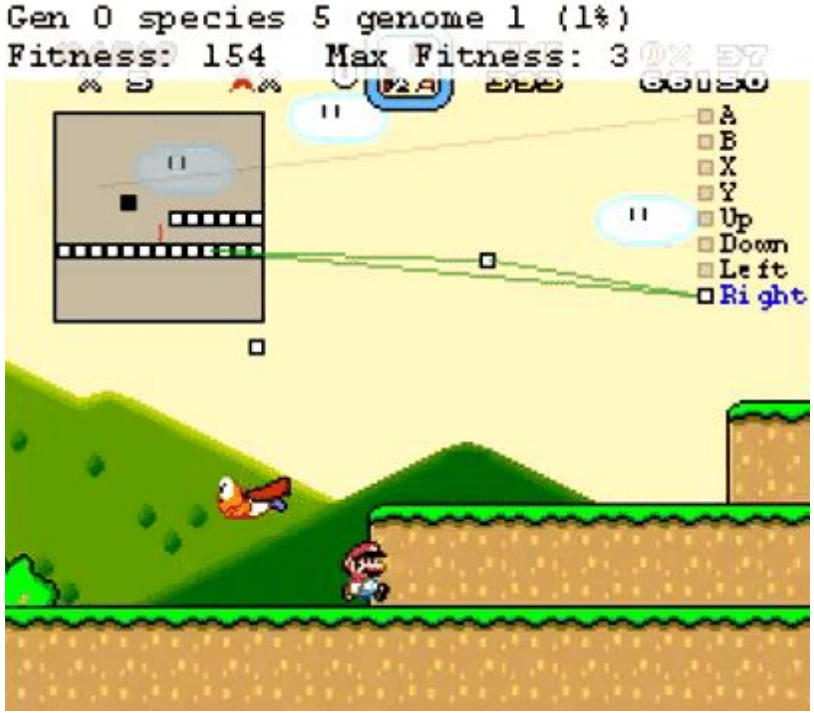
Source: http://bit.ly/2y6WpGb

Training MarI/O

MarI/O's fitness increases the further right MarI/O moves

Green lines represent a positive connection meaning continue the current input.

Red lines represent a negative connection meaning change the current input.



Source: http://bit.ly/2y6WpGb

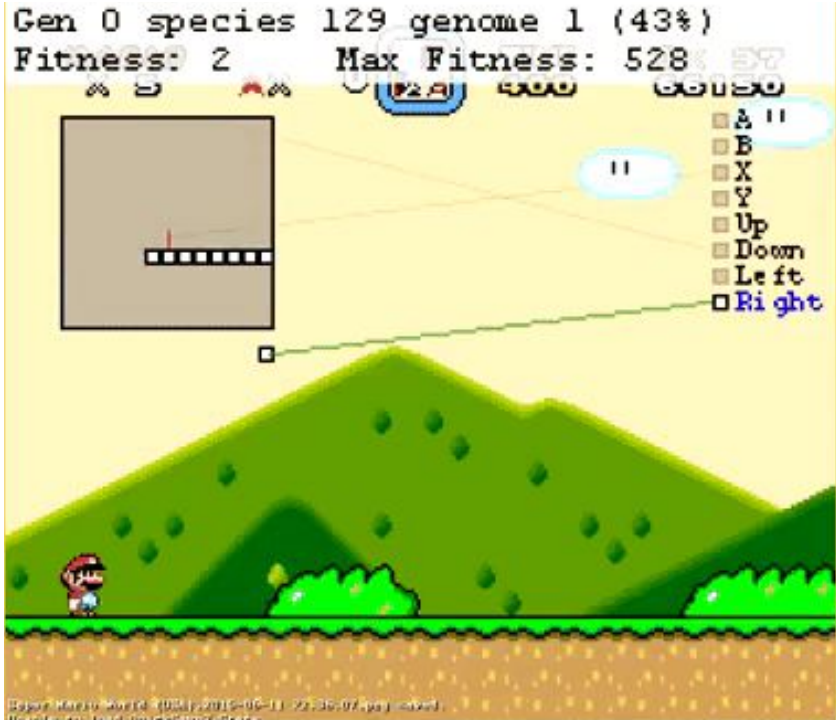
How MarI/O learns

N.E.A.T - Neuro Evolution of Augmenting Topologies

Choosing fittests of a given generation

Breeding them together

Adding random mutations



Source: http://bit.ly/2y6WpGb

Thank you

Q & A