



Reinforcement Learning in Games and Entertainment

Taken from: <https://cinnamonai.medium.com/reinforcement-learning-in-games-and-entertainment-47ffb22b9b9e>

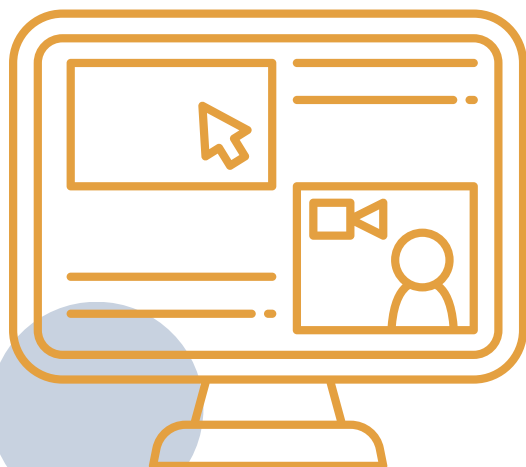
Reinforcement Learning in Games and Entertainment

The term “Reinforcement Learning” has been gaining popularity in recent years. As one of the branches of machine learning, it is understandable that it also enjoys the attention given to machine learning as a whole. But what exactly is reinforcement learning? And where does it shine compared to other machine learning solutions? In this article, we will examine reinforcement learning from the perspective of real-world problems and introduce an example application in the Games and Entertainment industry.



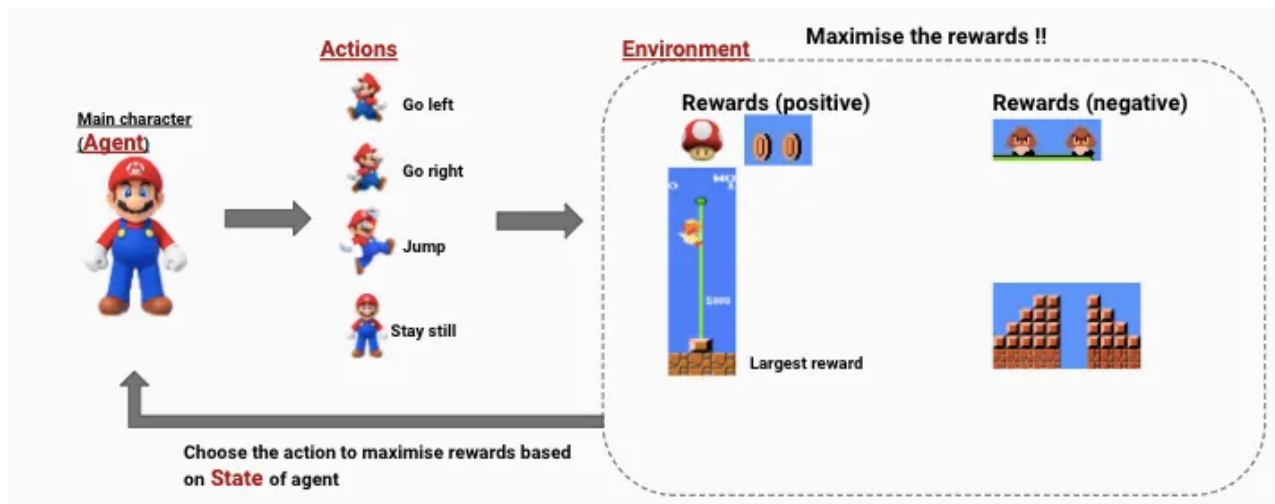
Modeling real-world problems

When we consider actual tasks performed by humans in the real world, we can define some descriptors. In any task, there will always be an Agent, which is the human performing the task, a Goal of the task, and a Reward or Penalty for completing or failing the task. In order to complete the task, the Agent needs to perform a series of Actions on the Environment which contains external forces that either help or obstruct the Agent while keeping track of its State.



For example, in the task of food delivery, the rider is the agent, and he/she needs to navigate the streets (actions) to reach the goal of reaching the customer’s house while also making sure he/she is on the correct route (state). The biggest reward comes when the rider reaches the customer and delivers the food, and penalties in between can come in the form of taking a wrong turn, caught in a traffic jam, etc. that prevents the rider from completing the task.

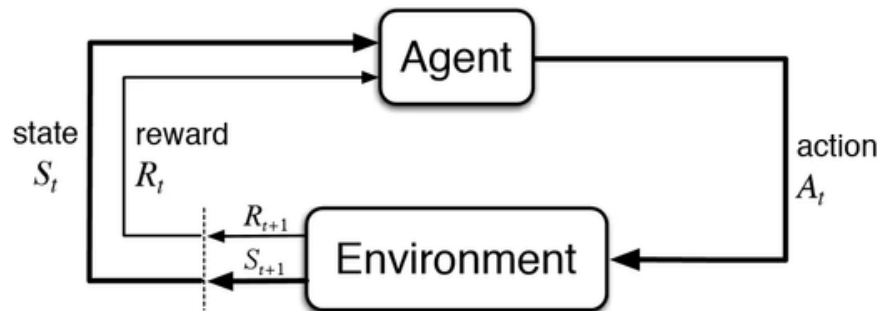
In order to better understand the above definitions in games, let's examine a simple example. In the game Mario Brothers, the character needs to achieve certain goals by performing a predefined set of actions. The character needs to reach the flag, collecting coins and items on the way and also avoiding enemies and pits (penalties).



It just so happens that the formal mathematical definition of reinforcement learning is as follows:

1. At t_0 , the Agent does not know what Action to take so it can take a random Action, or other strategies if there is preliminary knowledge.
2. At a timestep t_t , the Agent performs an Action (A_t)
3. At the next timestep ($t_t + 1$) the Agent perceives its new State (S_{t+1}) and also considers the Rewards (R_{t+1}) it got from the Environment as the result of its Action.
4. If the Rewards get smaller then the Agent will adjust to choose a different Action
5. This process is repeated until the Agent completes running its episode





Reinforcement Learning applications in the game industry

From the examples above, it is easy to notice that this definition of reinforcement learning fits perfectly with many real-world problems, and because games are often based on real-world scenarios, games become a perfect application of reinforcement learning. We have shown an example of an application in a simple game in the previous section and here we will see how it is used in the gaming industry.

Automated game testing



Imagine a game not so simple like the Mario example, but an FPS like the screenshot above. There are almost infinite patterns of Actions that the Agent can take. Walking, running, shooting, changing weapons type, reloading, interacting with objects and items, and the list goes on.

Conventional game testing normally includes tens of hours of gameplay by beta testers to make sure the game works fine, not just free from coding bugs, but also specifications bugs such as a ledge that is too high for the player to jump on, etc. By automating gameplay, testers just need to set a certain time to let the AI complete the game, and if the AI cannot complete the game, then only testers will be required to check the logs for why it couldn't complete.

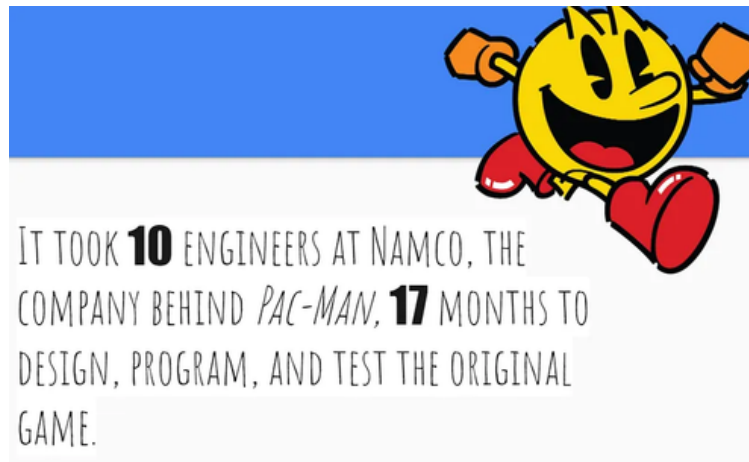


Automated game tuning

Another common usage of reinforcement learning is in NPC or enemy behavior tuning. For example, in a fighting game developers normally want to create various characters with different skills and abilities but overall evenly matched. Reinforcement learning can help by automating the fight between AI characters and taking the win/lose statistics. If the statistics are heavily skewed towards a certain character, the developers can adjust that character's abilities and damage dealt. All these testings can be done automatically so it will remove a greater amount of burden from testers.



If only these techniques were available many years ago, we would have seen more games released every year.



Industry and market conditions

Unfortunately, deploying Reinforcement learning on actual game development workflows have quite considerable challenges, and even more to create a common platform or best practices. At the moment, although most game companies are starting to develop some kind of machine learning solutions in-house, there are no major players who provide Reinforcement learning services to the game industry. We believe the following factors are the greatest obstacles to widely implement a common Reinforcement learning platform/pipeline in the game industry.

Challenges in implementing Reinforcement learning

- **Reinforcement learning behavior is unpredictable**

This applies to all fields in machine learning, where complexity has come to the point that it is impossible for humans to understand completely why an AI agent behaves in a certain way. For game designers who want to provide a constant gaming experience, this becomes a barrier to implement AI on game characters.



- **Reinforcement learning architecture design is coupled with the game design**

The Actions that the Agent can take are different across games, genres, and companies. And of course, the Environment will also greatly influence the design of Rewards. Because of these limitations, to design a Reinforcement learning architecture we will need access to the game designs and source codes as well. This fact can present a number of problems but the biggest one is that for most game companies, their code and assets are their most valuable Intellectual property assets. Many game companies are very reluctant to share the codes and would choose to contain all development in-house.



- **Reinforcement learning requires heavy computation during inference**

Only games running on high-powered consoles or PCs can afford to run a reinforcement learning-based game. Although the usage varies, it is safe to assume that it would be better if the device has at least a decent GPU. This prevents reinforcement learning to be implemented widely since many casual gamers are playing games on their smartphones that are either low spec or too battery draining to be running reinforcement learning inferences. This also requires game companies to invest in parallel programming and various optimizations.

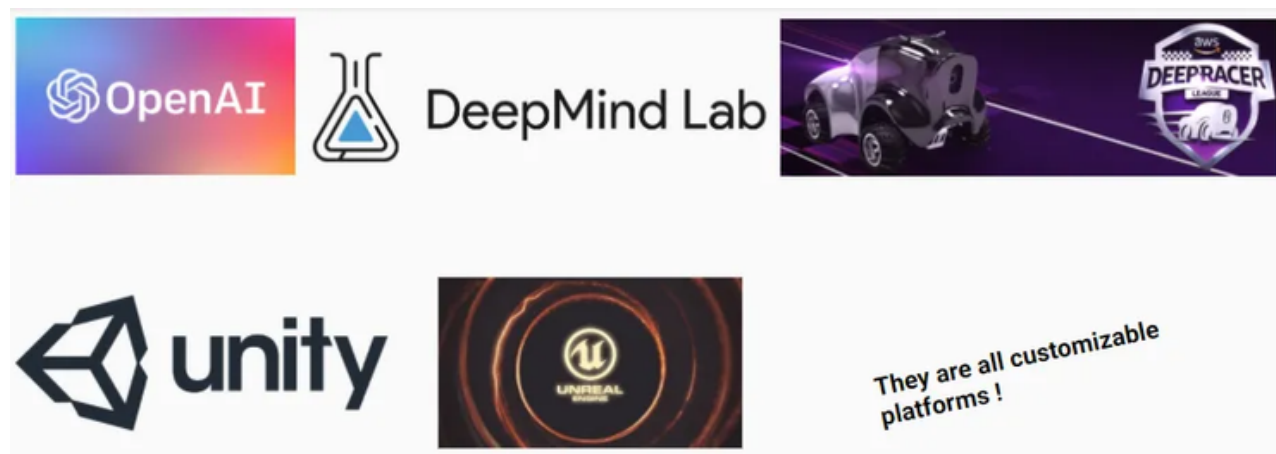


What gaming companies can do to mitigate the unpredictability of reinforcement learning

Game developers can apply reinforcement learning to non-critical aspects of the game. Enemy behavior for the most common low-level enemies, testing but only on limited areas of the game, etc. are good candidates for limited implementation of reinforcement learning.

What non-gaming companies are doing to spur the growth of Reinforcement learning

Since game companies are reluctant to share code and knowledge, many companies have started providing tools and platforms to help other companies build their own reinforcement learning systems.



OpenAI, DeepMind Lab are notable providers of Reinforcement learning tools and platforms that enable engineers to develop and test customized Reinforcement learning strategies. While game engine manufacturers such as Unity and Unreal also provide easy to integrate Reinforcement learning libraries to their respective software. These initiatives have helped to break into Reinforcement learning a lot easier especially for game developers.

Recap

In this article, we discussed the basics of reinforcement learning and how it mimics most real-world problems. Then we moved on to look at some applications of reinforcement learning in the gaming industry and the challenges of implementing it in-game development workflows. Finally, we mentioned some ways to mitigate the challenges and described the situation of reinforcement learning adoption in the industry.

