

Solving a skill labyrinth using Reinforcement Learning with Unity ML-Agents and Curriculum Learning

Mario da Graca

Self-optimizing Systems

Winter Term 2022/23

University of Applied Sciences (HAW Hamburg)

Berliner Tor 5, 20999 Hamburg

Abstract. This project report examines curriculum learning in Unity ML-Agents using a 3D skill labyrinth as an example. The task involves tilting the board on two axes to guide a marble through a labyrinth, filled with walls and holes, through which the agent must steer the marble in order to follow the correct path and reach the end. The agent's performance was evaluated using two different approaches: standard reinforcement learning and curriculum learning. Curriculum learning involves starting the agent with simpler sub-tasks, and gradually increasing the difficulty over time. The paper covers the technical details of the project, including setup of the environment, tuning of parameters, reward engineering and performance analysis of the trained agent.

Keywords: Reinforcement Learning · Curriculum Learning · Unity ML-Agents

1 Introduction

Machine learning (ML) is a subset of Artificial Intelligence (AI) that focuses on the development of algorithms and statistical models that allow computers to learn from data and make predictions or judgments without being explicitly programmed. Reinforcement Learning is a type of Machine Learning that focuses on teaching agents to make decisions in a given environment by maximizing a reward signal. It is a method for an agent to learn how to behave in a given situation by executing specific actions and watching the rewards that the environment provides.

ML and RL are widely used in a range of applications, such as robotics, autonomous vehicles, finance, healthcare and games. This report explores Reinforcement Learning in the latter one, by creating a complex and agility focused labyrinth environment, that an agent has to successfully solve. With the help of Unity's ML-Agents [2] a model is trained using the standard Reinforcement Learning and Curriculum Learning [3] approach. The goal of this project was to explore the process of an agent learning a fine motor task. The following chapter

will explain the setup of the board game in real life and as a game in the Unity engine [1] to lay the grounds for the third chapter, where the reinforcement learning aspects of this project are described. The focus is on the environment and training setup, as well as the reward engineering. In chapter four the used metrics are discussed and the results presented, in order to give a qualified statement in chapter five about the performances and show where the process can be improved.

2 Game Implementation

2.1 Game board

The basic setup of the game is simple, yet the hard parts lie in the way the game is played. As shown in Fig. 1 the game board is made out of wood and tilt-able on two disconnected axes. This allows the player to move the marble around without direct contact, just by leveraging gravitation. The goal of this game is to reach the end on the right hand side of the board, without falling into a hole. If the marble falls down, the player has to restart and is rewarded with the points that are associated with each hole. To follow the rules, the marble has to pass the holes in the correct order given by the black line. Only if these requirements are met, the player can win the game. There is no time limit in this game.

The basic concepts are easy to learn, but solving the task is hard to master. The game requires fine adjustments to the angles and quick reactions to prevent the marble from rolling uncontrolled through the labyrinth.



Fig. 1. The wooden game board, with the path to follow. Source: [4]

2.2 Unity game board

To closely approximate the movement and behavior of the marble in this physics based skill game, Unity’s built-in physics engine (PhysX3). This system provides several components that can be attached to game objects, to ensure specific behaviors. A game object in Unity defines the base for any entity in the given scene.

1. Collider3D: The Collider3D component can be added to any game object, either static or dynamic, to simulate the contact points of hard surfaces, like the marble or the game board. The shape is usually defined by or similar to the actual 3D model. This component can serve two purposes. It either behaves like a collider mentioned above or, if specified by a flag, only acts as a trigger, meaning it’s not colliding with other Colliders, but still sends out a signal, if other Collider3D game objects collide with this game object.
2. Rigidbody3D: To add game objects to the physics simulation that are affected by gravity, the Rigidbody3D component with the associated flag can be added to game objects with a Collider3D.

References

1. Haas, J.K.: A history of the unity game engine (2014)
2. Juliani, A., Berges, V.P., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., Lange, D.: Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627 (2020)
3. Narvekar, S., Stone, P.: Learning Curriculum Policies for Reinforcement Learning (2018). <https://doi.org/10.48550/ARXIV.1812.00285>, <https://arxiv.org/abs/1812.00285>, publisher: arXiv Version Number: 1
4. unknown: Wooden labyrinth puzzle maze game 36 pass balance board educational skill improvement toys for kids (unknown), <https://shopee.com.my/Wooden-Labyrinth-Puzzle-Maze-Game-36-Pass-Balance-Board-Educational-Skill-Improvement-Toys-for-Kids-i.82073268.2508203260>, [Online; accessed January 30, 2023]