FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Una Adilović

Playing a 3D Tunnel Game Using Reinforcement Learning

Department of Software and Computer Science Education (KSVI)

Supervisor of the bachelor thesis: Adam Dingle, M.Sc.
Study programme: Computer Science
Study branch: Artificial Intelligence

Prague 2023

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to express my heartfelt appreciation to my supervisor, Adam Dingle, for his invaluable guidance and support throughout my academic journey, beginning with our first Programming 1 class and continuing through to the completion of this project. His patience and assistance were greatly appreciated and played a significant role in the success of this work.

In addition, I am deeply thankful to my mother for her unwavering belief in me and to my grandparents who made it possible for me to be here. Their contributions are greatly appreciated and will never be forgotten.

Title: Playing a 3D Tunnel Game Using Reinforcement Learning

Author: Una Adilović

Department: Department of Software and Computer Science Education (KSVI)

Supervisor: Adam Dingle, M.Sc., Department of Software and Computer Science Education (KSVI)

Abstract: Tunnel games are a type of 3D video game in which the player moves through a tunnel and tries to avoid obstacles by rotating around the axis of the tunnel. These games often involve fast-paced gameplay and require quick reflexes and spatial awareness to navigate through the tunnel successfully. The aim of this thesis is to explore the representation of a tunnel game in a discrete manner and to compare various reinforcement learning algorithms in this context. The objective is to evaluate the performance of these algorithms in a game setting and identify potential strengths and limitations. The results of this study may offer insights on the application of discrete tabular methods in the development of AI agents for other continuous games. (ADD RESULT)

Keywords: tunnel game, reinforcement learning, artificial intelligence, algorithms

Contents

Introduction

Reinforcement learning is a subfield of machine learning that aims to train agents to make decisions that will maximize a reward signal. This approach has been widely applied in the field of artificial intelligence, particularly in the context of training agents to play games. In a game setting, an agent's actions can be evaluated based on their impact on the agent's score or likelihood of winning. Through the process of reinforcement learning, the agent learns to make strategic decisions that maximize its reward by receiving positive reinforcement for good moves and negative reinforcement for suboptimal moves. This allows the agent to adapt and improve its performance over time as it plays the game. Research on reinforcement learning in games has demonstrated its effectiveness in a variety of contexts, including board games, video games, and real-time strategy games.

In the field of artificial intelligence, games can be classified as either continuous or discrete based on the nature of the action space and state space. Continuous games have a continuous action space, meaning that the possible actions an agent can take are not limited to a fixed set of options, but can vary continuously within a certain range. In contrast, discrete games have a discrete action space, meaning that the possible actions are limited to a fixed set of options. Continuous games are often characterized by a high-dimensional state space, as they may involve a large number of variables that describe the game state. Discrete games, on the other hand, typically have a lower-dimensional state space, as the number of possible states is limited by the discrete action space. In general, continuous games are more challenging to model and solve than discrete games, as they require more complex decision-making algorithms and may require more computational resources.

In this thesis, we will investigate the application of reinforcement learning to train agents to play a continuous 3D tunnel game. The continuous game environment will be discretized into a set of states, and different reinforcement learning algorithms will be applied to train agents to play the game. The goal of this study is to determine whether it is possible for any of the agents to win the whole game, and to compare the performance of different agents that use different reinforcement learning algorithms.

The results of this study will contribute to the understanding of the potential of reinforcement learning for training agents to use discreate algorithms in a naturally continuous environment, and to provide insight into the strengths and weaknesses of different reinforcement learning algorithms in this context.

1. Game Design

The game that will be analysed is called ”Space-run” and it involves attempting to accumulate the highest score possible by navigating through three distinct tunnels while avoiding various obstacles. The game is endless in nature, as the speed increases each time the player successfully completes all three tunnels.

1.1 Player and Movement

In ”Space-run” the player assumes control of a character named Hans, who is responsible for dodging obstacles by moving left and right. In addition to these lateral movements, Hans also has the ability to shoot bullets and defeat certain in-game creatures, which results in a higher score. The player must utilize these abilities in order to progress through the game and achieve a high score.

1.2 Obstacles

As previously stated, the player must navigate through various obstacles in the game. These obstacles can be divided into three distinct categories, and each tunnel contains a unique subset of them. In the subsequent sections, we will delve deeper into these categories in order to better understand the challenges faced by the player.

1.2.1 Traps

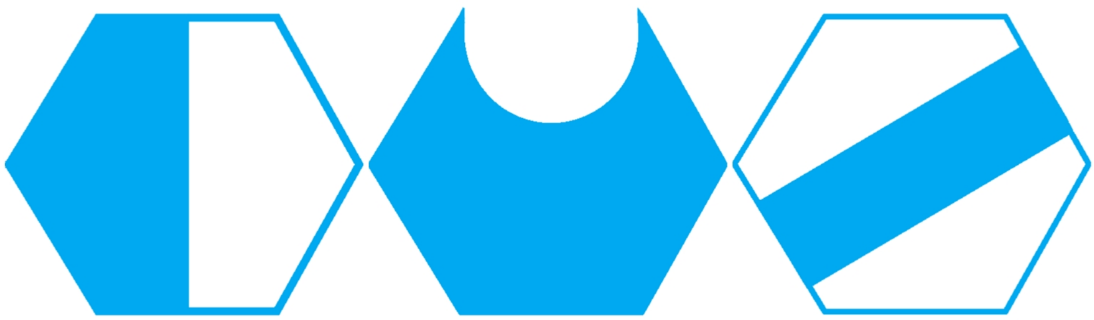

Figure 1.1: Trap examples

As depicted in Figure 1.1, a selection of the various trap types that the character Hans must avoid is presented. These traps, of which there are a total of 10, vary in their level of difficulty and can be either static or animated. These traps can be encountered in any of the three tunnels, and if the player fails to successfully evade them, they result in an instant death.

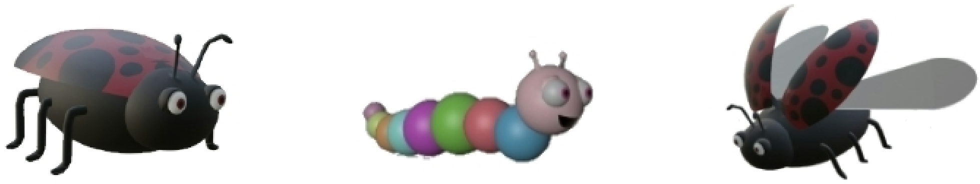

Figure 1.2: Bugs

1.2.2 Bugs

In addition to traps, the game also features bugs as an obstacle (as shown in Figure 1.2). These bugs typically appear in the second tunnel and are designed to align with the player, making them more challenging to evade. However, they can still be avoided by the player. If the player chooses to engage with the bugs, they can be defeated by shooting three bullets at them. If the player collides with a bug, Hans will lose 25% of his battery life(for more information on battery life, see Section 1.3).

1.2.3 Viruses

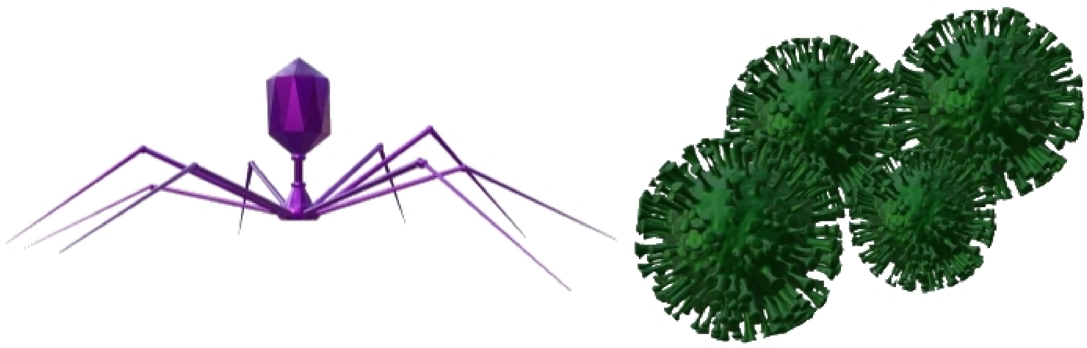

Figure 1.3: Bacteriophage and Rotavirus

The third and final type of obstacle in the game are viruses (illustrated in Figure 1.3). These viruses are typically found in the third tunnel and, like bugs, can be eliminated through the use of three bullets. They also align with the player's movement, making them more difficult to evade, although it is still possible to do so. Bacteriophage, a subtype of virus, will result in an instant death if the player comes into contact with them. Rotaviruses, on the other hand, will cause the player's character to become sick for a brief period of time. During this illness, it is crucial for the player to avoid coming into contact with another Rotavirus, as this will result in the end of the game.

1.3 Additional Features

There are several other features of the game that are worth mentioning. One of the most significant of these is the battery life of the player's character, Hans, which is displayed on the right side of the screen. As Hans is designed to resemble a computer, it is necessary for him to recharge his battery throughout the game by collecting energy tokens. This will fully restore his battery capacity. There are three main ways in which Hans can lose battery life: running causes a constant reduction of 1%, each bullet shot costs 1% of the battery life, and coming into contact with a bug results in a reduction of 25% (as described in Section 1.2.2). If the battery reaches 0%, Hans will die and the game will end.

Finally, it should be noted that upon successfully navigating through all three types of tunnels, the game will increase in speed and the player will once again encounter the same tunnels, looping through them indefinitely until the player loses.

1.4 Score Count and Winning

The score of the game is based on the length of time that the player is able to survive. Additionally, each time a player successfully shoots down a bug or virus, their score increases by 10 points. As previously mentioned, the game is designed to be played indefinitely, but for the purpose of this study, we have set the game to be considered won after an agent successfully completes nine tunnels, reaching level 10.

2. Implementation of the Game

intro about godot

in this chapter describe code structure (from isp)

2.1 Title of the first subchapter of the second chapter

2.2 Title of the second subchapter of the second chapter

3. Structure of the Experimental Setting

code structure of the environment, basically how main.gd works

3.1 Title of the first subchapter of the second chapter

3.2 Title of the second subchapter of the second chapter

4. Applied Algorithms

talk ab algos

4.1 Title of the first subchapter of the second chapter

4.2 Title of the second subchapter of the second chapter

5. Implementation of the Agents

code structure of the agents

5.1 Title of the first subchapter of the second chapter

5.2 Title of the second subchapter of the second chapter

6. Experiments

An example citation: Anděl [2007]

6.1 Title of the first subchapter of the second chapter

6.2 Title of the second subchapter of the second chapter

Conclusion

Bibliography

J. Anděl. *Základy matematické statistiky*. Druhé opravené vydání. Matfyzpress, Praha, 2007. ISBN 80-7378-001-1.

List of Figures

List of Tables

List of Abbreviations

A. Attachments

A.1 First Attachment