Tank War: A 2D AI Confrontation Game

Xi Zhu

Seidenberg School of Computer Science and Information Systems
1 Pace Plaza, New York, NY 10038, USA
xz79622p@pace.edu

***Abstract*— This article introduces Tank War - a 2D AI Confrontation Game. At present, there are many 2D Battle Games on the market. However, due to the problems of simple playability, uneven quality, and inoperability of traditional 2D Battle Games, it is difficult for players to obtain a satisfactory game experience. To solve these problems, we developed a 2D AI Confrontation Game that enhances the gameplay experience by introducing Machine Learning techniques to enhance gameplay and strategy. Players will get a whole new experience in this Game, different than traditional 2D AI Confrontation Games, by picking up in-game items, changing the custom look, fighting with escalating AI opponents, and developing different strategies against different levels. This Game is to provide a better game experience for the player.**

***Keywords*— Game Design, Game Development, Artificial Intelligence, Machine Learning, Reinforcement Learning**

I. INTRODUCTION

With the increasing demand for gameplay, a new type of combat game was created -- Tank Wars. In this 2D game, players control tanks to fight each other on a virtual battlefield. With the help of artificial intelligence technology, players can experience challenging combat scenarios. War of Tanks is expected to be a web game accessible via a website. The game offers a variety of maps and tanks to choose from, providing players with a unique and exciting gameplay experience. Tank Wars will be an exciting and fun game that challenges the player's strategic thinking and combat skills.

II. LITERATURE REVIEW

This section provides four similar games and analyzes their main functions and modes.

A. TANK MAYHEM

TANK MAYHEM is a fast-paced online game where players control a tank to battle enemies. Furthermore, the game allows players to play local battles. In the game, players can pick up various items with different functions to upgrade their tanks and thus pose a threat to their opponents' tanks. The game features different maps and generated items for each round, which can keep players engaged for hours. However, the game's handling may disadvantage some players, limiting its long-term appeal.

B. Clash of Tanks

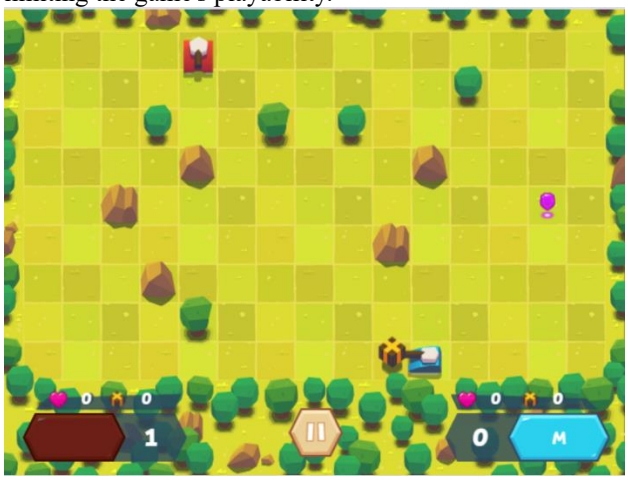
Clash of Tanks is a very strategic game, but the most significant difference is that players need to deploy different kinds of tanks rather than control them. The player aims to destroy the opponent's base by deploying various types of upgradable tanks. The game's main strength is its novel gameplay and the choice of multiple tanks, allowing players to choose their own strategy and play style. However, the game's difficulty is also a disadvantage for some players who prefer a more relaxed experience.

C. TANKS: SCI-FI BATTLE

TANKS: SCI-FI BATTLE is a top-down tank battle game with rich visual effects. The player's goal is to use his intelligence to destroy all opponents. The main advantage of the game is its strategy, which requires the player to take advantage of the fact that projectiles bounce off walls to destroy the opponent behind cover. However, the difficulty of the game is also a disadvantage for some players who prefer a more relaxed experience.

D. Tank Battle (2020)

Tank Battle (2020) is a one-button tank duelling game that offers both player-vs-AI and player-vs-player gameplay modes. The game has a beautiful painting style and a variety of props. Players must upgrade their tanks by picking up items to defeat opponents. The game's main feature is that it only takes one button to control the tank, but it also limits the player's maneuverability as the tank can only turn right. Therefore, this is the biggest factor limiting the game's playability.



III. ARCHITECTURE DIAGRAM

A. Conceptual Diagram

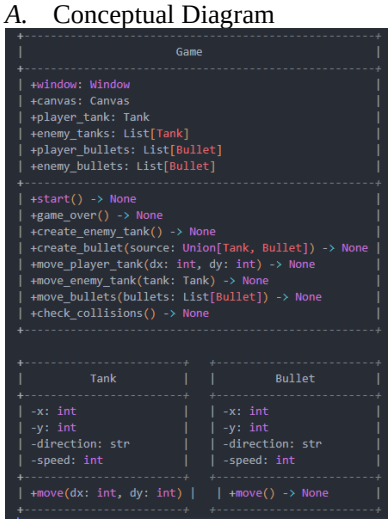


Figure 4.1 Conceptual Diagram

Figure 4.1 shows Conceptual Diagram. In the Game frame is a game object with a Window and a Canvas, Player tanks and Enemy tanks, player bullets, and Enemy Bullet. Game objects also contain many methods associated with game logic, such as creating enemy tanks, creating create_bullets, and checking check_collisions.

Player and enemy tanks have similar attributes, such as position (x, y), direction, and speed. They all have a move method to move their position.

Player Bullet and Enemy Bullet have similar properties, such as position (x, y), direction, and speed. They all have a move method to move their position.

As the diagram shows, the relationships between the components in this architecture are apparent. The Game object (Game) owns all the other components and is responsible for calling their methods to implement the game logic. There is also a clear relationship between player tanks, enemy tanks, player bullets, and enemy bullets.

B. Sequence Diagram

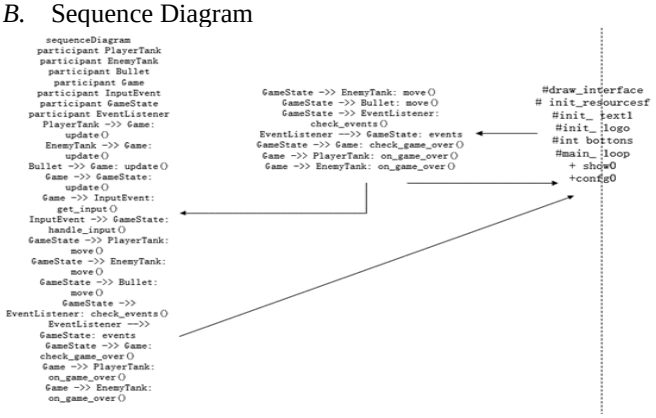


Figure 4.2 Sequence Diagram

Figure 4.2 shows Sequence Diagram. This sequence diagram shows the major actors and their interactions. When the player presses the button, the player tank object updates its direction and status. Player tank objects and enemy tank objects are then updated with their position and orientation on the game canvas. The enemy tank object creates new bullet objects and adds them to the enemy bullet object list. Similarly, player tank objects create new bullet objects and add them to the player's bullet object list. The bullet object checks its status and position and checks for a collision with an enemy tank or player tank. The game canvas object updates health and score.

C. Class Diagram

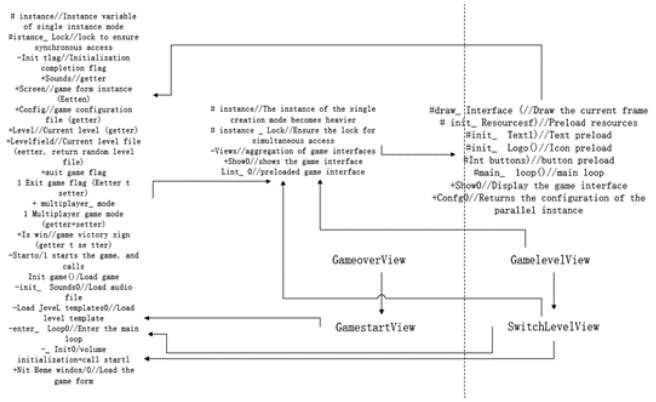


Figure 4.3 Class Diagram

Figure 4.3 Show Class Diagram. The Class Diagram of Tank War describes the various classes in the game and their relationships. In this game, the following classes can be identified:

GameWindow: This class represents the main window of the game, including the start, pause, continue, and end of the game.

Canvas class: This class represents the canvas in the game on which all game objects are drawn.

Tanks: This class represents tanks in the game, both player and enemy, that can fire bullets and move.

Bullet: This class represents bullets in the game, including bullets from player and enemy tanks, which can be moved and collision detected.

Wall Class: This class represents walls in the game and allows collision detection.

Sound: This class represents the sound effects in the game, including explosions and shooting.

Menu: This class represents the menu in the game, including options to start the game, save the game, and load the game.

The relationship between them is as follows:

The GameWindow contains a Canvas object on which all game objects are drawn. The Canvas contains multiple Tank, Bullet, and Wall objects, which can be drawn and moved on the canvas. The Tank class contains multiple Bullet objects that fire bullets and move them. The Bullet class can detect movement and collision, and if it collides with a Wall or Tank, it will produce an explosion and play a Sound. The Menu class interacts with the GameWindow and Canvas, providing options to start the game, quit the game, and so on.

D. Use case Diagram

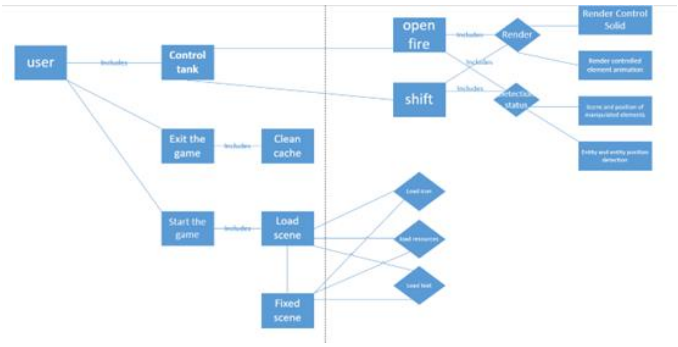


Figure 4.4 Use case Diagram

Figure 4.4 shows Use case Diagram. The Use Case Diagram is a UML (Unified Modeling Language) graphical representation that describes the function of the system and the interaction between users, and how the system meets user requirements. In a Tank war game, the following are possible use cases:

Start the game: The user starts the game by clicking the "Start Game" button.

Control tank movement: The user can control the movement of the player tank through the keyboard, including up, down, left, and right four directions.

Control tank firing: The user can control the player tank firing bullets from the keyboard.

Kill enemy tanks: The user wins the game by operating the player tank and killing enemy tanks.

In the above use case, the primary interaction between the user and the system is to control tank movement and firing, kill enemy tanks, and end the game depending on whether the game wins or loses. At the same time, the system also provides the function of saving games and loading games, allowing users to save the game state when needed and continue the game later. The Use Case Diagram provides a high-level view so that users can better understand the functionality of the system and the interactions between users.

IV. METHODOLOGY

This project is based on PyTorch, a Python-based machine learning library. PyTorch can be regarded as the encapsulation of numpy. Based on PyTorch, features required by machine learning are added, such as reverse gradient calculation, training data management, learning rate strategy, loss function, GPU operation, and distributed management. Therefore, Pytorch can quickly complete the construction and verification of machine learning tasks.

Reinforcement Learning (RL) is a branch of machine learning that realizes supervised learning through interaction with the environment to obtain feedback. Reinforcement learning includes learnable agents (namely neural networks) and environmental components. The agent obtains the observation state S from the environment and calculates it to obtain behavior A. After applying A to the environment, it can obtain the corresponding reward R (R can be positive or negative feedback) from the environment and enter the next observation state S '. If this process is repeated, A large amount of training data will be obtained, and offline learning will be carried out. The learning process is to input S into the neural network and obtain behavior A, which will be optimized according to the feedback of this behavior and the quality of the following state S '. It is equivalent to recalling and summarizing what happened and optimizing the original decision A to obtain better feedback R in the future.

In this project, the observation state S includes the map information of the tank game, the information of the NPC tank itself, and the information of the player tank, a matrix of three channels in total. There are five types of behavior A: up, down, left, right, and fire. When it is impossible to move up, behavior

A equals "up" will get negative feedback; When the NPC tank and the player tank are in the same row or column, and the NPC tank is facing the player tank, action A equals "fire" will get positive feedback. In addition, more environmental feedback strategies can be designed.

V. RESULTS

Reinforcement learning is still supervised learning, which requires data collection, and then learning in small batches by randomly using some samples. After more than three hours of essential learning (which is slow to train the model on the CPU), after 100,000 iterations of the model, the NPC's tank was able to approach the player's tank and fire when it was ready to attack. No matter where the player tank and the NPC tank are on the map, the NPC tank will learn how to approach the tank. Due to the large number of maps in the game and the fact that the training of the model includes map information, on some maps, NPC tanks will ignore obstacles and fire, and move inefficiently.

VI. CONCLUSION

The use of reinforcement learning in video games has been very mature. This project confirms the feasibility of the deep Q network algorithm in tank battle games. By reasonably designing rewards or punishments for each behavior in different scenarios of the game, a random neural network can gradually master the rules of the game and learn to take the optimal behavior in different scenarios to get as much reward as possible and avoid punishment.

Reinforcement learning can also carry out more complex tasks. The tasks of this project are relatively not complicated, and there is a lot of room for optimization, such as allowing NPC tanks to trigger items and change their attributes; Add different obstacles to bring a more affluent entertainment nature. By reinforcement learning, NPC tanks can learn more rules of the game.

Plans are to use reinforcement learning to train more extensive neural networks and migrate training tasks to GPUs for faster verification. Later on, the input features will be changed to the game screen, which makes learning more difficult, but makes more sense because the game screen is the only state information in the game.

REFERENCES

[1] "Clash of Tanks Game · play online for free · gamaverse.com," Clash of Tanks Game · Play Online For Free · Gamaverse.com. [Online]. Available: https://gamaverse.com/clash-of-tanks-game/. [Accessed: 09-Mar-2023].

[2] "Tank battle (2020) game · play online for free · gamaverse.com," Tank Battle (2020) Game · Play Online For Free · Gamaverse.com. [Online]. Available: https://gamaverse.com/tank-battle-2020-game/. [Accessed: 09-Mar-2023].

[3] "Tank Mayhem Game · play online for free · gamaverse.com," Tank Mayhem Game · Play Online For Free · Gamaverse.com. [Online]. Available: https://gamaverse.com/tank-mayhem-game/. [Accessed: 09-Mar-2023].

[4] "Tanks: Sci-Fi Battle Game · play online for free · gamaverse.com," Tanks: Sci-fi Battle Game · Play Online For Free · Gamaverse.com. [Online]. Available: https://gamaverse.com/tanks-sci-fi-battle-game/. [Accessed: 09-Mar-2023].

[5] E. Martinez, "Riot games and deep reinforcement learning in gaming," Anyscale, 03-May-2023. [Online]. Available: https://www.anyscale.com/blog/riot-games-and-deep-reinforcement-learning-in-gaming. [Accessed: 05-May-2023].

[6] Y. Feng, S. Subramanian, H. Wang, and S. Guo, "Train a mario-playing RL agent¶," Train a Mario-playing RL Agent - PyTorch Tutorials 2.0.0+cu117 documentation, 2023. [Online]. Available: https://pytorch.org/tutorials/intermediate/mario_rl_tutorial.html#train-a-mario-playing-rl-agent. [Accessed: 05-May-2023].

[7] H. Panchal, "Introduction to reinforcement learning (RL) in Pytorch," Medium, 06-Sep-2021. [Online]. Available: https://medium.com/analytics-vidhya/introduction-to-reinforcement-learning-rl-in-pytorch-c0862989cc0e. [Accessed: 05-May-2023].

[8] J. M. Carew, "What is reinforcement learning?: Definition from TechTarget," Enterprise AI, 10-Feb-2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning. [Accessed: 06-May-2023].