**Deep Q-Learning for Rapid Roll**

Bera Shi (bshi) Zhecheng Wang (zhecheng) Leon Li (yangli95)

**I. Motivation**

Reinforcement learning is a hot topic in machine learning field, in which we concern about how software agents make decision and take actions in an environment to maximize some notion of cumulative rewards. Nowadays, combination of deep learning and reinforcement learning becomes a trend in Artificial Intelligence. The implementation of Deep Reinforcement Learning (DRL) on video games will visually present how intelligent a machine can be, and to what extent machine can perform better than humans on definite tasks. In this project, we seek a single agent trained with DRL which can outperform human in playing a popular video game, Rapid Roll.

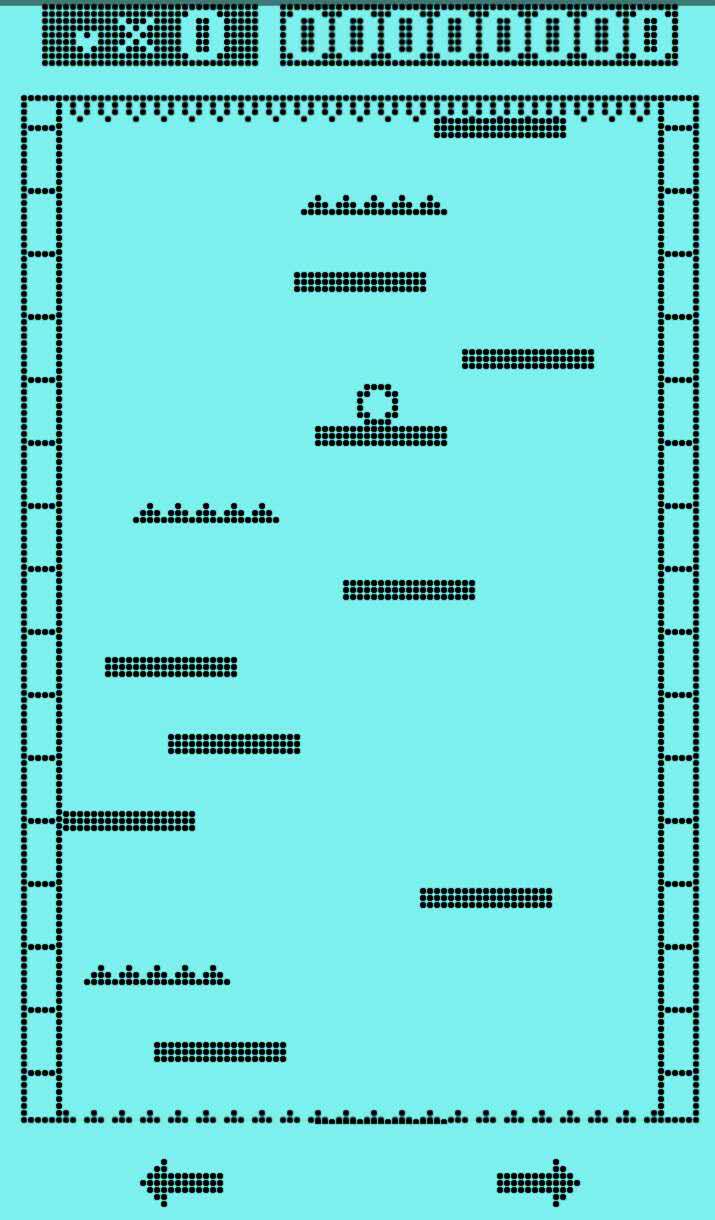


Figure 1: The screenshot of Rapid Roll game

A player is required to keep the ball going downstairs with all platforms rolling upwards in Rapid Roll game. The ball dies if it drops down directly or is killed by the spikes at the top or spike platforms. The player can move the ball left or right to jump from one platform to another. “Food” rewards may appear on some platform, providing additional life. The rule is very simple, but it is not easy for humans to keep alive for even seconds! We expect our agent to beat human players on this game!

**II. Related Work**

The project *Deep Reinforcement Learning for Flappy Bird*[1] by Kevin Chen trained an agent to play Flappy Bird game by implementing deep reinforcement learning. The author used a feature extractor and built a Deep Q-network to deal with the task. By learning screenshot pixels, the author has found optimal policy and achieved results exceeding human level. Compared with this project, our model has more output states/actions and deals with more complicated inputs.

**III. Methodology**

We plan to implement Deep Q-Learning on an agent and make it “learn” to play Rapid Roll. The input of the learning system will be the screenshots at every moment, without “telling” the machine where platforms are and where spikes are, etc. The output of the learning system will be the key state and action of current time: “left”, “right” or neither. We can decompose the problem into several steps:

1. Implement Android App “Rapid Roll” on our PC via Android Monitor.
2. Develop algorithm to capture screenshots of the game interface at every moment during the game.
3. Design the rewards: “alive” reward is “+1”, “death” reward is “-1” and “food” reward is “+0.5”. Because we would rather the agent to live longer in the game instead of risking to eat food.
4. Design and train the Deep Q-network for sufficient iteration to get the optimal policy.
5. Compare the performance of the agent with humans.
6. Compare the performance of Deep Q-learning algorithm with other reinforcement learning methods if possible.

The main challenges of this project are high-dimension inputs, various rewards corresponding to different blocks and the robustness to different speeds and difficulties. We also try to apply deeper networks to our project for high-dimension inputs and more key states and actions.

**IV. Timetable**

1. Milestone 1: Build agent and develop screenshots algorithm. 11/6
2. Milestone 2: Develop the Deep Q-Network. 11/16
3. Milestone 3: Training and evaluation. 11/30
4. Milestone 4: Summary and extension works. 12/10

**References**

[1] http://cs229.stanford.edu/proj2015/362\_report.pdf