# ARTIFICIAL INTELLIGENCE TERM PROJECT REPORT

Mehmet Onur Erboy – 15011007

Bahattin Cihan Ünal – 18011603

## 1. Project Description

For this project, we tried to implement an agent which will play a pong game against a real opponent using reinforcement learning.

The project has been developed with python programming language.

For accomplishing this task, we created a pong game using pygame package and we trained an agent with Q-learning techniques.

The details are described at further chapters.

## 2. The Solution Logic and Development Process

Upon developing this project, we created a simple pong game using pygame package. In the game,there are two plates that can move in the y axis for playing. One of them is for player and another for our agent to play. There is also a ball moves with a constant speed with 45 degrees, changes its direction when it hits on the horizontal borders or the plates. If the ball hits the vertical borders, the player or the agent scores 1 point. (Depending on the side that our ball has been hit.)

After implementing a pong game, we implemented a Q-learning algorithm. In our algorithm the agent calculates the Q for the given state (ball_x, ball_y, opponent_y, dir_x, dir_y) score with an depth limited search. For every state in the depth limited search, we punish and reward our agent relatively the distance between the ball and our agent or if it has caught or couldn't catch the ball. When the search is done for the current state we reward our agent using the angle between the ball and agent(it should be close to 45 degrees) and how much the agent is close to the ball. We use those results to predict the agents next movement.

After calculating those Q function results for many states, we save our Q function using the pickle package for further usage. For a probability epsilon our agent takes some random action for exploring new actions.

In the development process, the geometric calculations of the game, the parameters for reward and punishment were really tricky and did keep us busy.

## 3. Comments about project

Optimum depth value for our depth limited search is 9 when we try an higher number we see that the game slows down. We tried many reward and punishment values. But the best ones were:

If the depth was reached in our search:

Punish the agent with value -dist(ball, agent)

If x axis of the ball >= x axis of the opponent plate:

Punish the agent by -5 if agent couldn't catch the ball

Give reward to the agent by 5 else.

When calculating the Q function:

Give reward to the agent by - 2 * dist(ball, agent) + 2 * abs(sin(2*angle)) // this sin function gives the numerically larger results when the angle is closer to the 45 or -45 degrees.

## 4. References

- Clear Code, Learning Pygame by making Pong - https://www.youtube.com/watch?v=Qf3-aDXG8q4&t=1s
- Mehmet Fatih AMASYALI, Artificial Intelligence lecture notes, ML-3 - https://drive.google.com/file/d/1vEGe9N2UuP42EFd6WZBM_MNFPlKNIlQE/view

## 5. Numerical Success

We made the agent play the game with an excellent opponent for some time(changes the ball direction every time when it comes players side)

The results were, agent did caught 1000 balls and couldn't catch 59 balls.

We can say that success = 1000/1059 = 0,944

## 6. Game Screenhot