

# Machine Learning Workshop

## Tic-Tac-Toe Project

Martin Mrugała  
Patryk Walczak  
Filip Szymczak  
Bartek Żyła  
Maciej Zalewski

*June 12, 2020*



---

# *Contents*

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Standard Tic-tac-toe . . . . .	1
1.2	Our implementation . . . . .	2
1.3	Machine Learning backgroud . . . . .	2
<b>2</b>	<b>Code</b>	<b>3</b>
<b>3</b>	<b>Bibliography</b>	<b>5</b>



# *Introduction*

---

## 1.1 Standard Tic-tac-toe

According to the definition in the Oxford Dictionary of English, Tic-tac-toe is a game in which two players seek to complete a row of either three noughts or three crosses drawn alternately in the spaces of a grid of nine squares.



Figure 1.1: A completed game of Tic-tac-toe<sup>1</sup>.

And as it turns out, there are 255 168 possible games. So this means that, with the help of the reinforcement learning, a bot may train to mastery.

---

<sup>1</sup>Source of the image: <https://en.wikipedia.org/wiki/Tic-tac-toe>

## 1.2 Our implementation

Our team decided to create a bot which would play this game but in an expanded version of it. The principal changes concern, inter alia:

- **The size of the grid could be infinite**

The number of squares is not boundless, of course. Due to the CPU, as well as the storage limitations, the infinity has to be simulated. We came up with two solutions. The first one is pretty straightforward, but a little preachy. The idea is to set the size in advance. The second one, in turn, assumes an increase in the size whenever the players get close to the edge.

- **Condition of winning**

The length of the sequence of X's or O's needed to win may have any value. Naturally, it has to satisfy the following condition:

$$0 < length < edgelenh$$

There is still the matter of the draw. In case of the constant size grid, the tie occurs, whenever there is no way for any player to win.

## 1.3 Machine Learning backgroud

---

## *Code*

---

---

```
class MyClass(Yourclass):  
    def __init__(self, my, yours):  
        bla = '5 1 2 3 4'  
    print bla
```

---





---

# *Bibliography*

---

