Politechnika Śląska

Katedra Grafiki, Wizji komputerowej i Systemów Cyfrowych

Academc year			Group	Section
2022/2023	SSI	**BIAI**	**N**	**M**
Supervisor:	mgr inż. Marcin Wierzchanowski		**Classes: (day, hour)**	
Section members:	**Patryk Rossa, Wojciech Palka**		**Tuesday**	
emails:	**patros601@student.polsl.pl, wojcpal243@student.polsl.pl**		**13:00 – 14:30**	

Raport

Subject:

AI Flappy Bird

Main assumptions:

1. Introduction

The aim of the project was to create a flappy bird game in which the bird overcomes obstacles on its own using a neural network. After starting the program, it creates the number of birds that is specified in the model. Then the birds try to avoid obstacles and on this basis they learn to pass them.

2. Analysis of the task

2.1 Technologies used

To create our application, we used Python and its libraries, because we decided that it would be best suited for this project. The first library we used is PyGame which helped us create the main part i.e. the game. PyGame is one of the most popular library for game development and its easy and fun to learn. Next we needed the library for machine learning. We chose NEAT, because it allows to create a neural network for our game in a simple and transparent way.

2.2 Other options

Other possible technology that would work well in this project is also Python library: Keras. This is a popular library that allows to create neural networks, but we did not use it, because we did not find a guide that help us use this technology in our project.

3. Internal and external specification of the software solution

3.1 Classes

a. Bird

Bird class is representing birds moving through the game. The object of this class is learning how to overcome obstacles. The class consists of the following elements:

- Constants:
  - IMGS – set of images representing all possible wing positions of the bird object – it simulates real birds' wings movement.
  - MAX_ROTATION – max rotation of the bird object
  - ROT_VEL – velocity of the rotation
  - ANIMATION_TIME – time of the animation of the bird object – all images of the bird object are shown during this value of time in seconds.
- Variables:
  - x – x position of the bird object on the window
  - y – y position of the bird object on the window
  - tilt – tilt of the bird
  - tick_count – number of moves that bird has done since last jump (using for the calculation of velocity – the higher tick_count, the higher velocity)
  - vel – velocity of the bird
  - height – height on which the bird is in the game
  - img_count – indicates on which image the bird currently is

        - img – current image of the bird
    - Methods:
        - __init__() - constructor of the object – sets all values
        - jump() – makes the bird jump
        - move() – moves the bird on the window, calculation velocity and setting new x and y position and tilt of the bird
        - draw() – draws the bird on the screen, checks what image should the bird be and drawing the correct one
        - get_mask() – using for getting bird object bounds (i.e. in order to check collision with pipes)

b. Pipe

Pipe class is representing pipes – obstacles that make the game challenging. The class consist of the following elements:

- Constants:
    - GAP – gap between top and bottom pipe in pixels
    - VEL – velocity of the pipe (screen is moving forward to the right, so pipes must move to the left with some velocity)
- Variables:
    - x - x position of the pipe on the window
    - height – height of the pipe (should be random to make the game challenging)
    - top – y position of the top of the pipe
    - bottom – y position of the bottom of the pipe
    - PIPE_TOP – image of the top oriented pipe
    - PIPE_BOTTOM – image of the bottom oriented pipe
    - passed – Boolean value representing if birds has surpassed the pipe
- Methods:
    - __init__() – constructor of the object – sets all values
    - set_height() – calculates the height of the pipe
    - move() – moves the pipe on the left
    - draw() – draws the pipe on the screen
    - collide() – checks if given bird collide with the pipe

c. Base

Base class is representing another obstacle – the ground. The class consists of the following elements:

- Constants:
    - VEL – velocity of the base – analogically to the pipe
    - WIDTH – width of the base object
    - IMG – image of the base object
- Variables:
    - y – y position of the base on the screen
    - x1 – x position of the left boundary of the base on the screen
    - x2 – x position of the right boundary of the base on the screen
- Methods:
    - __init__() – contructor of the object – sets all values
    - move() – moves the base to the left (analogically to the pipe objects)
    - draw() – draws the base on the screen

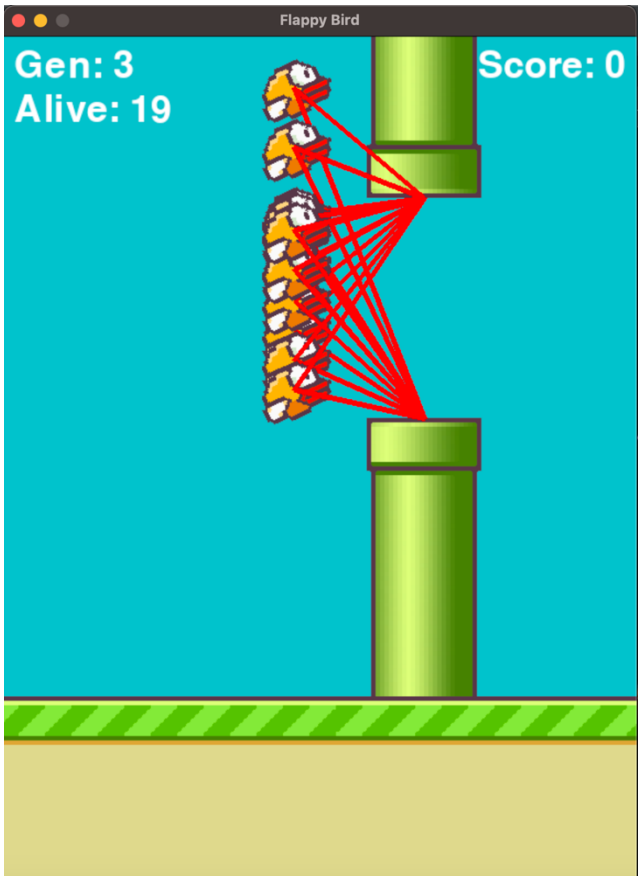
3.2 Functions

a. main() – initializes all of the objects that are needed to learn birds. Creates nets and genomes and manages them. Controls global variables that are displayed on the screen such as score (obstacles surpassed). And importantly – makes the pipes, base and birds move based on information returned by nets.

b. run() – gets configuration of the nets from the file and runs main function

3.3 User interface

The interface is a simulation of the game. Instead of playing by ourselves the nets are making decision whether make the bird jump or not.



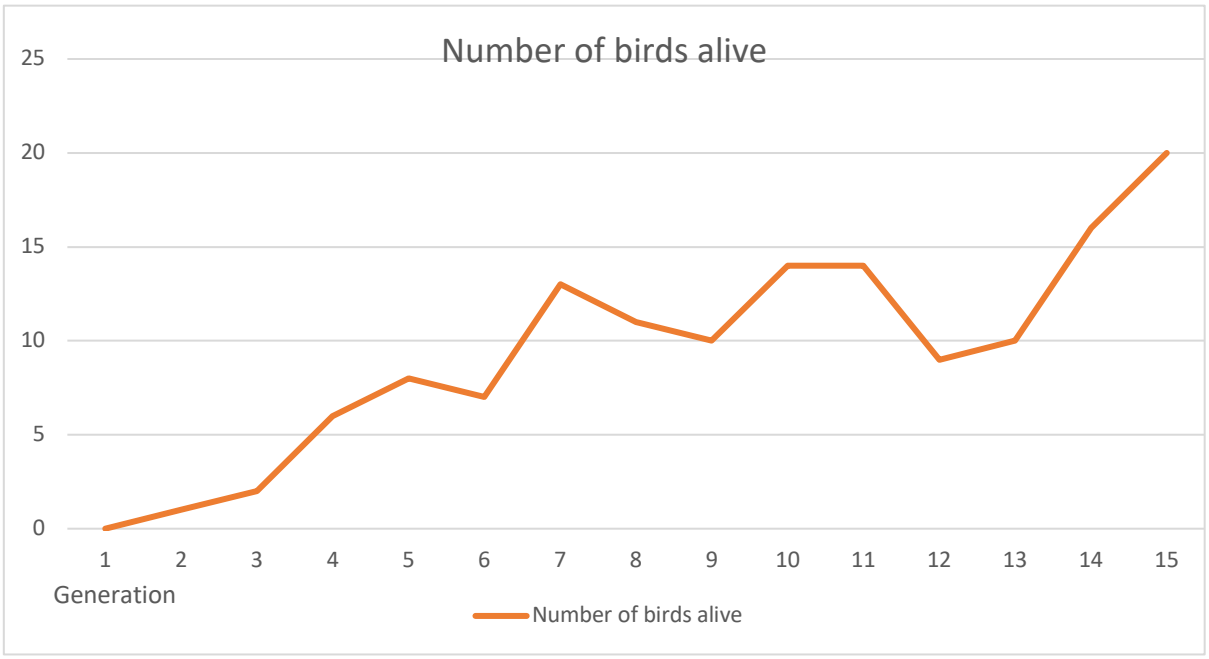
We can see number of active generation and number of alive birds on the top left corner and score (number of overpassed obstacles) on the top right corner. All of the gameplay is done by AI. We can easily observe the effects of learning when the number of alive birds and final score increases with every generation of birds.

4. Experiments

4.1 Experimental background

We have set the starting number of birds to 50. It seems for us to be an optimal number. In order to speed up the testing we have limited time of the gameplay to 5 pipes. It allows us to observe how many birds survived all 5 pipes and see how much the next generations are better from the previous ones. We have been also manipulating the fitness score of each bird to find out what is the optimal number in order to maximize the learning effects.

4.2 Presentation of experiments



The chart shows results of average number of birds alive during first 15 generations. As we can see the general tendency is increasing number of birds alive at 5th pipe. There are some fluctuations in chart line, but it is normal in learning process.

5. Conclusion

To sum up the project was enjoyable to do. We had the opportunity to learn the basics of artificial intelligence and delve into the technologies needed for this. We learned how to make a simple game in Python and how to create a neural network that will be used in the game.

6. References:

- https://docs.python.org/3/ - python docs
- https://www.pygame.org/docs/ - pygame docs
- https://neat-python.readthedocs.io/en/latest/ - neat docs