|  |  |
| --- | --- |
|  | **Wydział Finansów i Zarządzania** **Kierunek: Informatyka** |

Dawid Miśkiewicz

(numer albumu: 81794)

Temat pracy

The Comparison of Evolutionary Algorithms, Genetic Algorithms, and AI in Flappy Bird Development

Opiekun merytoryczny:

Mgr inż., Sebastian Bień

Wrocław 2024

Table of content

Introduction

1. Flappy Bird Game  
1.1 Overview of Flappy Bird Game Development

1.1.1 Tools and technologies used for development (pygame)

1.2 Designing the Game Architecture

1.2.1 Structuring the game logic (game loop, event handling, physics).

1.2.2 Creating sprites and animations (player, obstacles, background).

1.3 Implementing Game Physics

1.3.1 Simulating gravity and player movement.

1.3.2 Collision detection between the bird and pipes.

2. Evolutionary Algorithms

2.1 What it is

2.1.1 Definition and inspiration from natural selection.

2.1.2 Key components: population, fitness, selection, crossover, mutation.

2.2. How does it work

2.2.1 Initialization of the population.

2.2.2 Evaluation of individuals using a fitness function.

2.2.3 Selection of parents for reproduction.

2.2.4 Generating offspring through crossover and mutation.

2.3 Example of use

2.3.1 Solving combinatorial problems (e.g., traveling salesman problem).

2.3.2 Evolving game-playing agents.

2.4 Advantages and Limitations

3. Genetic Algorithms

3.1 What it is

3.1.1 A type of evolutionary algorithm based on genetics and natural selection.

3.1.2 Representation of solutions as chromosomes (bit strings, real numbers, etc.).

3.2. How does it work

3.2.1 Encoding solutions into a genetic format.

3.2.2 Fitness evaluation and ranking of individuals.

3.2.3 Genetic operators: selection (e.g., roulette wheel, tournament), crossover, mutation.

3.3 Example of use

3.3.1 Feature selection in machine learning.

3.3.2 Evolving strategies for board games.

3.3.3 Parameter optimization in engineering problems.

4. Comparison of Algorithms  
4.1 Evolutionary Algorithms in Flappy Bird  
4.2 Genetic Algorithms in Flappy Bird  
4.3 AI Approaches in Flappy Bird  
4.4 Strengths and Weaknesses of Each Approach

5. Results and Analysis  
5.1 Performance Comparisons  
5.2 Optimization Efficiency  
5.3 Game Performance Evaluation

CONCLUSIONS

BIBLIOGRAPHY

TABLE OF FIGURES

SUMMARY

Introduction

1. Flappy Bird Game

This chapter provides an overview of the development of the Flappy Bird game, highlighting the tools and technologies implemented, the design and implementation of the game architecture, and the core mechanics that govern its functionality.

1.1 Flappy Bird Game – basic information

Flappy Bird is a game developed in 2013 by a Vietnamese programmer Dong Nguyen under his company DOTGEARS COMPANY LIMITED. The game was released in May 2013 on the IOS platform and in January 2014 on the Android platform. At the start of 2014, the game gained an unexpected spike in popularity, leading to mixed reviews caused of its difficulty. The objective of the game is to pass through pipes with a **sprite** using only one button to jump. The player’s score is determined by the amount of pipes passed.

1.2 Tools and technologies used for development

The app was developed in the programming language Python using the game engine Pygame in an integrated development environment Visual Studio Code. The following technologies were used to achieve the goals set in the project:

* Python 3.11.9
* Pygame 2.4.0
* neat-python 0.92

1.3 Structuring the game logic – Creating sprites and animations

The main game loop is managed by main() and manual\_play() functions. Inside the loop, two steps occur. The first one – Event Handling is where the loop processes events such as window events for exiting the game and key presses for jumping the bird.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

The next step is physics which is handled by the Bird class in the update\_position() method, where gravity and jump mechanics are applied to the bird. The bird’s position is updated for each frame based on velocity that is adjusted with gravity. Pipes move with constant speed and their height is random, as defined in the Pipe class.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

1.3.1 Player (Bird):

The bird is represented using three images (in flight) that have been loaded using pygame.image.load() and scaled to fit inside the window. Animation is controlled in the animation() method inside the Bird class, which shows different bird images according to a counter.



Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

The bird is then drawn with a rotation to simulate tilting during movement. The rotation is handled by pygame.transform.rotate().

1.3.2 Obstacles (Pipes):

Pipes are represented by an image (pipe.png) and have a top and a bottom section. The Pipe class creates, moves, and draws these pipes. Pipes are portrayed through the animation() method of the Pipe class.

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

Pipes move across the screen in the update\_position() method by decreasing their x position and new pipes are periodically created.

1.3.3 Background and Base:

The background (BG.png) and the base (base.png) are drawn in the draw\_window() function. The base’s horizontal movement creates a scrolling effect, and when it moves out of the screen, it reappears on the opposite side. This is done by the Base class:

Obraz zawierający zrzut ekranu, woda

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, żółty, Prostokąt, zieleń

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1.4 Implementing Game Physics

1.4.1 Simulating gravity and player movement   
 In the game, the motion of the bird is governed by a simplified kinematic equation. Said bird’s motion is handled in the PlayerBird class, where gravity and upward movement are defined by the equation presented below. The gravitational effect on the bird’s vertical position (y) is calculated as:

Where:

is the bird’s initial upward velocity of the bird. It is set to -10.5 units when the bird jumps.

is a constant downward gravitational acceleration. The acceleration is approximated to 3 units /

it the time step, tracked as tick\_count.

The position change is calculated as:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

To guarantee smooth motion and prevent excessive falling speed, the velocity is capped at a maximum of 16 units:



The bird’s vertical position (self.y) is then updated by adding to its current value:  


When the bird jumps, its velocity is set to an upward value, resisting gravity temporarily. The jump() function resets the parameters to simulate the jump effect:

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

The bird tilt is also adjusted to reflect its vertical movement visually. When the bird is rising, the tilt is positive, simulating a nose-up effect. Thus, the tilt takes on a positive value during ascension, creating a nose-up impression, and negative during descent, tilting gently downwards until reaching :

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

This guarantees that it appears realistic concerning the effect of gravity in the game.

1.4.2 Collision detection between the bird and pipes

The collision detection mechanism provides an accurate detection between the player (bird) and the game obstacles (pipes). Pixel-perfect collision is achieved by using masks to give a binary representation of the visible pixels of game objects.

1.4.2.1 Mask Creation

Masks are generated for the bird as well as both the bottom and top pipes.

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

1.4.2.2 Mask Offsets

Offsets are calculated to cancel offsets from the bird’s mask to the pipe's masks. The offsets are taking into account the relative positions of the bird and pipes:

Obraz zawierający zrzut ekranu, Czcionka, tekst, linia

Opis wygenerowany automatycznie

1.4.2.3 Overlap Detection

The overlap() method is used to determine whether any pixels in the bird’s mask overlap with the pipe masks:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

If any overlap is detected, a collision has occurred:

Obraz zawierający tekst, Czcionka, zrzut ekranu, numer

Opis wygenerowany automatycznie

This method ensures that collisions are accurately detected, even with tricky shapes such as a tilted bird.

2. Evolutionary Algorithm

The part concerned with operational principles and design principles of the application developed for this particular project will be presented in this chapter. It aims to go into the methodology and implementation issues, focusing on the key components with interrelations and, further still, discussing practical examples of use cases.

2.1 What is it?

An evolutionary algorithm is a computer-based approach to problem-solving modeled on natural evolution through the application of computer science and artificial intelligence. It simulates certain biological mechanisms, namely reproduction, mutation, and recombination, in order to solve these problems. Such algorithms are based on Darwin's conception of natural selection whereby weaker solutions are sorted from stronger ones and the more fit alternatives are preserved and bred, generation after generation. The ultimate goal is the development of solutions that are optimal enough to fulfill the desired purpose.

2.1.1 Definition and inspiration from natural selection