

DWIT COLLEGE
DEERWALK INSTITUTE OF TECHNOLOGY



AUTO PLAYING FLAPPY BIRD

A MINI PROJECT REPORT

Submitted to
Department of Computer Science
DWIT College

Submitted by
Pranjal Pandey
31St October, 2019

DWIT College
DEERWALK INSTITUTE OF TECHNOLOGY

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by PRANJAL PANDEY entitled “**AUTO PLAYING FLAPPY BIRD**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

.....

Dr. Sunil Chaudhary
HOD, Department of Computer Science
Deerwalk Institute of Technology
DWIT College

DWIT College
DEERWALK INSTITUTE OF TECHNOLOGY

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than that listed here have been used in this work.

.....

Pranjal Pandey

31st October, 2019

DWIT College
DEERWALK INSTITUTE OF TECHNOLOGY

LETTER OF APPROVAL

This is to certify that this project prepared by PRANJAL PANDEY entitled “**AUTO PLAYING FLAPPY BIRD**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

<p>.....</p> <p style="text-align: center;">Dr. Sunil Chaudhary HOD, Department of Computer Science Deerwalk Institute of Technology</p>	<p>.....</p> <p style="text-align: center;">Mr. Ritu Raj Lamsal HOD, Department of Electronics Deerwalk Institute of Technology</p>
--	---

ACKNOWLEDGEMENT

I would like to thank Dr. Sunil Chaudhary, HOD, Department of Computer Science at DWIT College for supervising this project and providing assistance when faced with problems. Without his guidance and persistent help, this project would not have been possible.

I am highly indebted to Deerwalk Institute of Technology for their guidance and constant Supervision as well as for providing necessary information regarding the project and also for their support in completing the project

I would also like to thank my parents and family, who have always supported me and to all of my friends who directly or indirectly helped me to complete this project and its report.

Pranjal Pandey

Roll No.:- 635

31st October, 2019

ABSTRACT

The application is named “Auto Playing Flappy Bird” since it is a demonstration where an AI (bird) learns to play the original Flappy Bird game and tries to master it. The game is a demonstration of Genetic Algorithm (Based on Darwin’s Theory of Evolution). It shows how a bird without any prior information about its surroundings slowly and gradually learns to adapt itself in it. Not only adapt, but even learns to pass the genes to its offspring making them even fittest for the environment. The bird adapt itself to such an extent that it could completely outperform the game and play it forever. Learning the steps of evolution, progressing slowly and gradually, the bird masters the game, the same game which had to be removed from the app store for being “much difficult” for humans. The application is built in JavaScript programming language using the p5 library from the same language. The application runs on any device with a simple http server and a modern web browsers installed in it. The end product is a fully trained flappy bird that is able to play the game forever and only fails when the game is stopped forcefully.

Keywords: *Auto Playing; Reinforcement Learning; Genetic Algorithm; Outperform Humans; Theory of Evolution; Survival of the Fittest*

TABLE OF CONTENTS

ACKNOWLEDGEMENT	iv
ABSTRACT.....	v
TABLE OF CONTENTS	vi
LIST OF FIGURES.....	viii
LIST OF TABLE	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1: INTRODUCTION	1
1.1. OVERVIEW.....	1
1.2. BACKGROUND AND MOTIVATION	1
1.3. PROBLEM STATEMENT	2
1.4. OBJECTIVE.....	3
1.5. SCOPE	3
1.6. OUTLINE.....	3
CHAPTER 2: BACKGROUND RESEARCH	5
2.1. LITERATURE REVIEW.....	5
2.1.1 Flappy Bird	5
2.2. CURRENT SYSTEM	6
2.3. THE PROBLEM WITH CURRENT SYSTEM.....	6
CHAPTER 3: SPECIFICATION AND DESIGN	7
3.1. REQUIREMENT ELICITATION AND ANALYSIS	7
3.1.1. Functional Requirement.....	7
3.2.2. Non-Functional Requirement.....	7
3.3. SYSTEM DESIGN	8
3.3.1. Use Case Diagram	8
3.3.2 Flow-Chart	9
CHAPTER 4: IMPLEMENTATION AND EVALUATION.....	10
4.1. TOOL AND TECHNOLOGY.....	10
4.2. IMPLEMENTATION	10

4.3. EVALUATION AND RESULT.....	13
CHAPTER 5: CONCLUSION	14
CHAPTER 6: LIMITATION.....	15
REFERENCES	16
APPENDIX I: Reinforcement Learning.....	17
APPENDIX II: Master the Game.....	18

LIST OF FIGURES

Figure 1: Use Case Diagram.....	6
Figure 2: Flow Chart.....	7

LIST OF TABLE

Table 1: Outline of Document.....	3
-----------------------------------	---

LIST OF ABBREVIATIONS

DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
2D	Two Dimension
AI	Artificial Intelligence
GA	Genetic Algorithm
PC	Personal Computer
HTTP	Hypertext Transfer Protocol

CHAPTER 1: INTRODUCTION

1.1. OVERVIEW

Auto Playing Flappy Bird is a demonstration of genetic algorithm and reinforcement learning where the bird trains itself to play the original Flappy Bird game which was a side-scrolling mobile game featuring 2D retro style graphics. The objective is to direct a flying bird who moves continuously to the right, between sets of Mario-like [2] pipes. If the bird touches the pipes, it dies and game stops. The bird briefly flaps upward each time that the player taps the screen; if the screen is not tapped, it falls because of gravity; each pair of pipes that he navigates between earns the bird a single point.

There is no variation or evolution in gameplay throughout the game, as the pipes always have the same gap between them and there is no end to the running track.

1.2. BACKGROUND AND MOTIVATION

Flappy bird has always been criticized for being insanely irritating, difficult and frustrating game [3]. Later, the creators had to remove it from app store due to its frustratingly addictive nature.

So, I wanted to build an AI that would train itself to such a level that it would outperform humans and the game itself. I thought it would be some kind of achievement to be able to train an AI to a level that it could outperform humans on a problem created by humans.

Besides, I have always been fascinated by Charles Darwin's theory of evolution [4]. And the algorithm I have used to train the bird works on the same principle of evolution. Genetic Algorithm [5] simulates the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate "survival of the fittest" among individual of consecutive generation for solving a problem. To portray the concept of evolution through an AI that evolves to such level that it outperforms the game have always been my dream. I had been searching a problem that best suits the algorithm to be portrayed and thought it wouldn't be any better to test the algorithm in one of the hardest games developed, so hard that had to be deleted from the app store for its difficulty and frustrating nature.

1.3. PROBLEM STATEMENT

The major reason of controversy for the original flappy bird game was due to its high level of difficulty, plagiarism in graphics and game mechanics [6]. It was accused of copying from the super Mario game. Through this project, I wanted to create a model that could play one of the hardest and most criticized games ever built by humans. Not just simply play, but outperform the game itself. To tackle the plagiarism, the bird, pipes and the background were completely changed without hampering the game's working mechanism.

1.4. OBJECTIVE

Using the Genetic algorithm and Reinforcement learning, this project trains an AI (bird) to automatically jump over the pipes (obstacles) and outperform the game. The bird should score highest score possible and should continuously play until someone interrupts.

1.5. SCOPE

Auto playing Flappy Bird can be played by anyone with a PC with modern web browsers installed in it. A simple http server needs to be activated before running the application.

1.6. OUTLINE

This paper includes the method used to collect the functional requirement and system specification. This paper also includes the diagrams to illustrate the system architecture. Moreover, it provides information about the tools and technologies used to build the system. It also provides the information about how those tools and technologies are implemented. Finally, it includes the method used for system testing and system evaluation. The overall report is organized as follows.

INTRODUCTION	<ul style="list-style-type: none"> • Overview • Background and Motivations • Problem Statement • Objective • Scope • Outline
BACKGROUND AND RESEARCH	<ul style="list-style-type: none"> • Literature Review • Current System • Problem with the current system
SPECIFICATION AN DESIGN	<ul style="list-style-type: none"> • Requirement Elicitation and Analysis • System Design
IMPLEMENTATION AND EVALUATION	<ul style="list-style-type: none"> • Tools and Technology • Implementation • Evaluation and Result
CONCLUSION	<ul style="list-style-type: none"> • Conclusion
LIMITATION	<ul style="list-style-type: none"> • Limitation

Table 1: Outline of the Document

CHAPTER 2: BACKGROUND RESEARCH

2.1. LITERATURE REVIEW

2.1.1 Flappy Bird

Flappy Bird was a mobile game developed by Vietnamese video game artist and programmer Dong Nguyen, under his game development company dotGears [7]. The game was a side-scroller where the player controls a bird, attempting to fly between columns of green pipes without hitting them. The game was released in May 2013 but received a sudden rise in popularity in early 2014 [8]. Flappy Bird received poor reviews from some critics, who criticized its high level of difficulty, plagiarism in graphics and game mechanics, while other reviewers found it addictive. At the end of January 2014, it was the most downloaded free game in the App Store for iOS [9].

On February 8, 2014, Nguyen announced on Twitter that the game was to be removed from both Apple's App Store and Google Play following the criticism it had constantly been facing.

2.2. CURRENT SYSTEM

There are some applications that simulate the Flappy Bird game and train the bird to play the game automatically. However, all the applications I found didn't seem to use the Genetic Algorithm and Reinforcement learning to train the bird. Some other machine learning algorithms like backtracking methods were found to be used. Hence, my curiosity towards the genetic algorithm and none of the projects using it encouraged me even more to build this application.

2.3. THE PROBLEM WITH CURRENT SYSTEM

All the applications I found didn't seem to have very pleasing visuals. They all looked exactly the same like the original Flappy Bird game and worked just fine. But my application tries to master the game using different visuals and designs. Further, the previously built applications didn't guarantee the bird to outperform the game whereas mine does.

CHAPTER 3: SPECIFICATION AND DESIGN

3.1. REQUIREMENT ELICITATION AND ANALYSIS

In order to find out how the actual Flappy Bird game works and the physics behind the game, I visited <http://flappybird.io/> which simulates the actual game. I played for sometimes and noticed the working mechanism of the game and what should be done to implement Genetic Algorithm and train the bird to play the game by itself.

3.1.1. Functional Requirement

- The bird shall start a new generation each time it hits the pipe.
- The bird shall always move right on the screen.
- The bird shall increase its fitness level on each new generation.
- The bird shall try to obtain high score.
- The bird shall always try to prevent collision.

3.2.2. Non-Functional Requirement

- The bird must carefully move up and down determining the positions of the pipes.
- The score must increase in each successful miss.
- The background must constantly change as the game marches on.
- The gap between two randomly generated pipes must be same.
- A new generation must start each time the bird fails to progress.
- The bird must outperform the game and constantly play trying to obtain as much score as possible.

- There must not be any variations on the gameplay throughout the game.

3.3. SYSTEM DESIGN

Use case diagram and Sequence diagram in designed and displayed below. As my project does not contain any databases, design of DFD and ERD aren't quite feasible.

3.3.1. Use Case Diagram

There is only one actor in the system that is the bird itself. The bird trains itself to play the game following the instructions. The bird can start a new game, jump between the pipes and prevent collision, try to obtain high score as the game marches on. And finally, the bird must increase its fitness level after each generations and master the game.

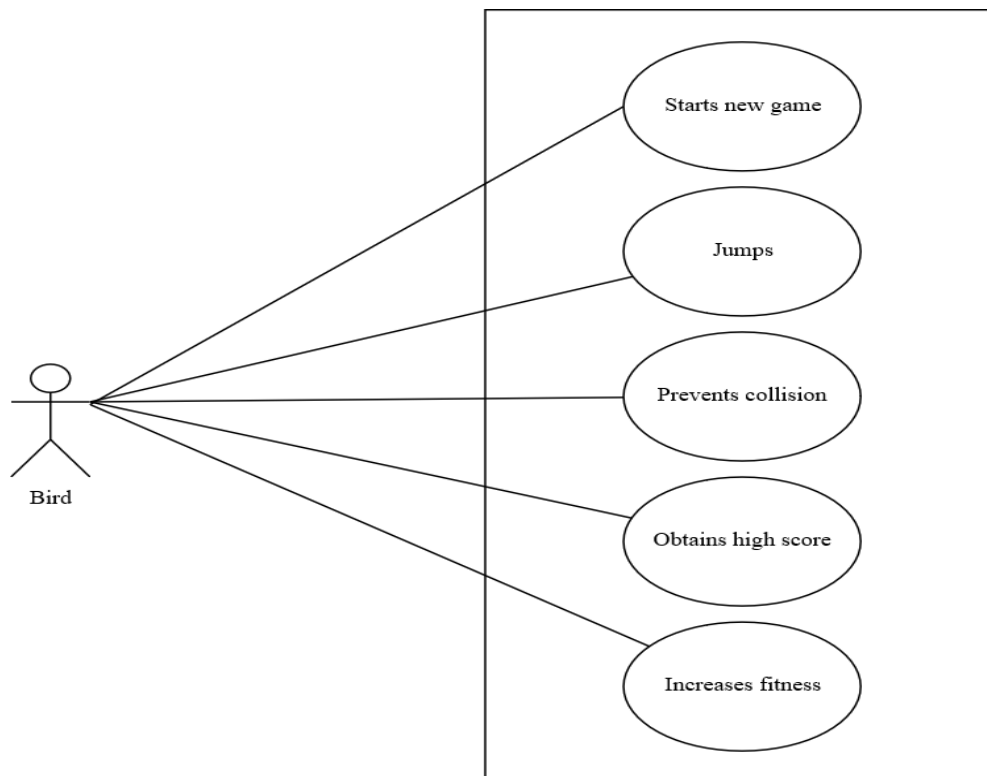


Figure 1: Use-Case Diagram

3.3.2 Flow-Chart

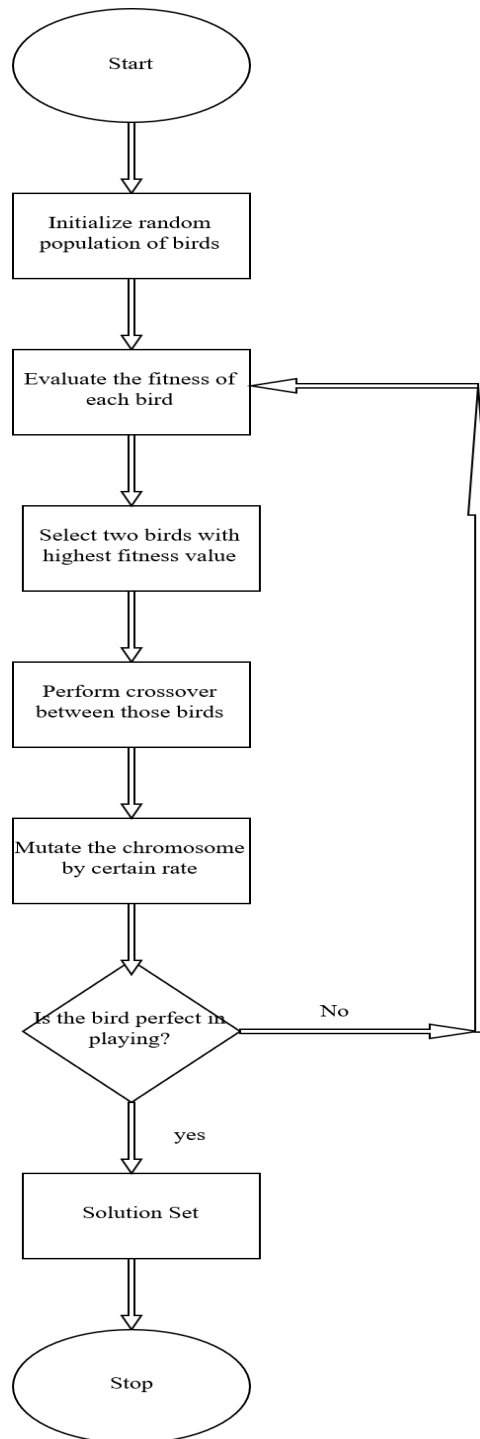


Figure 2: Flow-Chart

CHAPTER 4: IMPLEMENTATION AND EVALUATION

4.1. TOOL AND TECHNOLOGY

- System Specification and Design: Use case diagram, Genetic Algorithm, Reinforcement learning, System Flow diagram
- System Development: JavaScript programming language was used to develop the system. P5 library from the same language was used for visuals and canvas.
- System Testing: The system could be tested in a web browser running on a local server. A simple http server was needed to be activated for the program to run.

4.2. IMPLEMENTATION

The main goal of the application was to perform the functional requirement efficiently and Effectively. Also, another important goal was to make the visuals pleasing. I downloaded the background pictures and made my own design since I wanted to look my application different than the original one. For the visuals to look pleasing I used the p5 library from JavaScript. The library really made the designing part simple and effective. Now, that my design was completed, I had to implement the algorithm and made the bird play the game itself.

The algorithm that was used in the project can be summarized as:

- 1) Randomly initialize population p
- 2) Determine fitness of population

3) Until convergence repeat:

- a) Select parents from population
- b) Crossover and generate new population
- c) Perform mutation on new population
- d) Calculate fitness for new population

The above mentioned algorithm was implemented using the object oriented approach. Each tasks were divided into functions and the functions were called whenever necessary. The major functions used to implement the genetic algorithms were: copy, nextGeneration, calculateFitness, pickOne and mutate. The actual code for the genetic algorithm was implemented as:

```
function nextGeneration() {  
  console.log('next generation');  
  calculateFitness();  
  for (let i = 0; i < TOTAL; i++) {  
    birds[i] = pickOne();  
  }  
  savedBirds = [];  
}
```

```
function pickOne() {  
  let index = 0;  
  let r = random(1);  
  while (r > 0) {  
    r = r - savedBirds[index].fitness;  
    index++;  
  }  
  index--;
```

```
let bird = savedBirds[index];  
let child = new Bird(bird.brain);  
child.mutate();  
return child;  
}
```

```
function calculateFitness() {  
  let sum = 0;  
  for (let bird of savedBirds) {  
    sum += bird.score;  
  }  
  for (let bird of savedBirds) {  
    bird.fitness = bird.score / sum;  
  }  
}
```

Here, the function `calculateFitness()` calculates the fitness value of an individual bird and stores it in an array. The function `pickOne()` selects the bird with highest fitness level from the array and makes a copy of itself. After two birds with highest fitness are created, crossover is performed between them and the child produced from the process is supposed to have higher fitness than the parents. The process is repeated until a bird with enough fitness that is able to play the game flawlessly is produced.

4.3. EVALUATION AND RESULT

The game was tested in Web Browsers like Mozilla Firefox and Google Chrome that supported JavaScript. An http server was run on the device and the port 8000 was used. The test was successful as the game flawlessly ran and was up to the expectation. The bird was able to train itself to play the game and outperform it as well. Though it took the bird some generations to create a working model, it worked none the less. The bird was able to score more than 1000 points during its 30 minutes of testing period before it was closed while the game was still running and was thus able to outperform the game. The score the bird obtained was probably one of the, if not, the highest score I heard any human score in a Flappy Bird game.

CHAPTER 5: CONCLUSION

The main idea behind this project was to portray the concept of evolution. How living beings constantly evolve as the time passes, how they get used to their environment and evolve themselves to adapt to those environment. Evolution does not occur instantly. It takes some times and a lot of learnings to adapt to any environment. The large number of generations required for the birds to generate an offspring with highest fitness level portrays the same concept.

The objective of training an AI to such a level that it would outperform humans in a problem created by humans was some sort of success. The bird that was completely unaware about its environment and the game physics was able to successfully train itself to an extent that it could play the game forever. The score of more than 1000 supports my premise. The score the bird obtained was probably one of the, if not, the highest score I heard any human score in a Flappy Bird game.

CHAPTER 6: LIMITATION

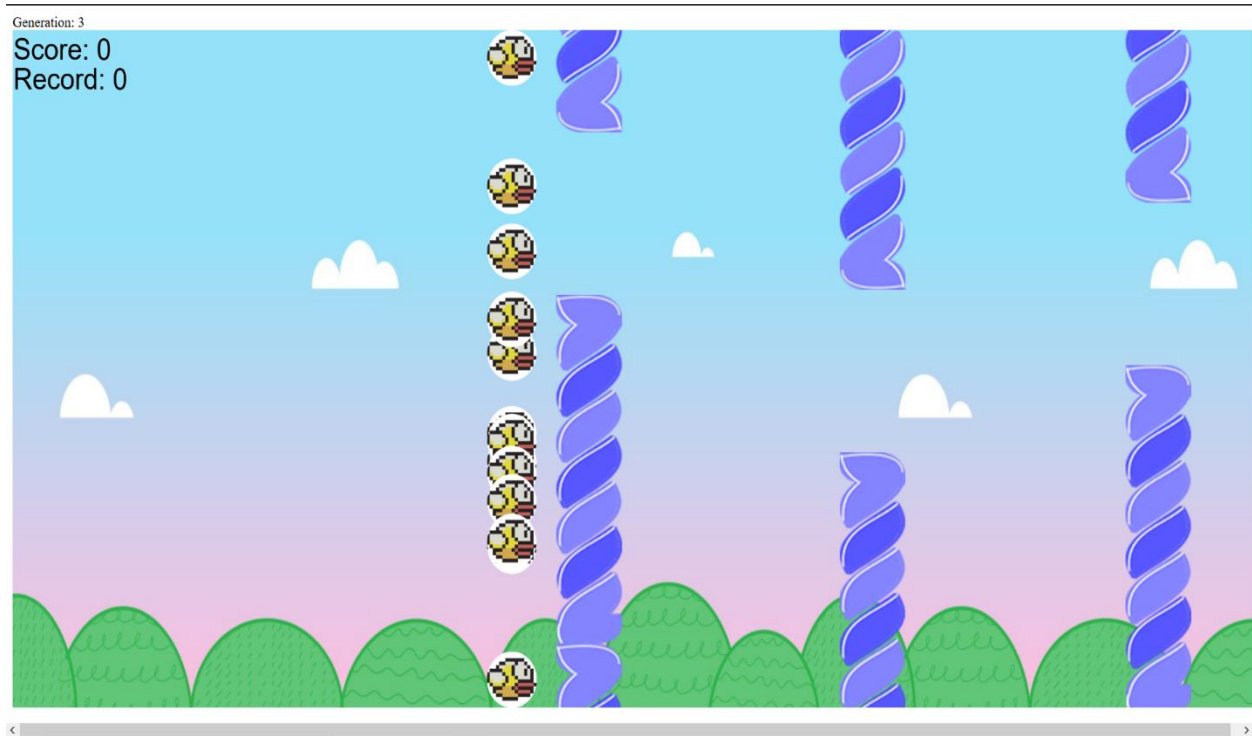
The major limitation of the application is that it is quite unpredictable to determine when exactly a fittest bird is produced. It requires many generations to obtain a fit population, hence the testing phase could be time consuming. Due to the limitations of the algorithm, we cannot predict the exact fitness of the bird at any given time. Further, the application is just a simulation of the algorithm and thus there is less interaction between human and the application. All user could do is to watch and wait an AI trains itself to master a game. An AI versus Human version of the same game would have been more appropriate and interesting.

REFERENCES

- [1, 4]"Darwin's Theory Of Evolution", Darwins-theory-of-evolution.com, 2019. [Online]. Available: <https://www.darwins-theory-of-evolution.com/>. [Accessed: 26- Oct- 2019].
- [2]"Play SUPER MARIO Games Free Online", Puffgames.com, 2019. [Online]. Available: http://www.puffgames.com/super_mario_games.html. [Accessed: 25- Oct- 2019]. *eBook*
- [3, 8, 9] Williams, R. (2019). What is Flappy Bird? The game taking the App Store by storm. [online] Telegraph.co.uk. Available at: <https://www.telegraph.co.uk/technology/news/10604366/What-is-Flappy-Bird-The-game-taking-the-App-Store-by-storm.html> [Accessed 28 Oct. 2019].
- [5]"Genetic Algorithms - GeeksforGeeks", GeeksforGeeks, 2019. [Online]. Available: <https://www.geeksforgeeks.org/genetic-algorithms/>. [Accessed: 26- Oct- 2019]. *Journal article*
- [6] Williams, R. (2019). What is Flappy Bird? The game taking the App Store by storm. [online] Telegraph.co.uk. Available at: <https://www.telegraph.co.uk/technology/news/10604366/What-is-Flappy-Bird-The-game-taking-the-App-Store-by-storm.html> [Accessed 30 Oct. 2019].
- [7]".GEARS | Game Studio", Dotgears.com, 2019. [Online]. Available: <http://dotgears.com/>. [Accessed: 30- Oct- 2019].

APPENDIX I: Reinforcement Learning

Here, randomly generated birds try to learn the game rule using the reinforcement learning method. They try out different moves and the move which leads to success are passed to the offspring in the next generation.



APPENDIX II: Master the Game

Here, we can see an evolved bird, evolved enough to master the game. The score of 655 before the game being stopped forcefully supports the claim.

