

# Final Project Proposal

Connor Holm

University of Minnesota- Twin Cities  
Minneapolis, Minnesota, USA  
holm0850@umn.edu

Jonathan Blixt

University of Minnesota- Twin Cities  
Minneapolis, Minnesota, USA  
blixt013@umn.edu

## 1 Introduction

Our project proposal revolves around the idea of using genetic algorithms to find a way to optimally play the popular game "Flappy Bird". In May of 2013, Flappy Bird was released and instantly became a hit. The game consists of a bird that is flying through a series of pipes until it either hits the pipes, the ground, or the sky. Through this project, we are going to find a way to code an agent that will be able to effectively navigate the game's obstacles.

## 2 Background

Genetic Algorithms are based on the idea of swapping genes in strands of DNA to simulate the process of natural selection. The idea of genetic algorithms stems from the work of Charles Darwin and his ideas of evolution and natural selection. Computer scientists took these ideas and used them in the field of AI. [2] Inside a genetic algorithm, there are two main ways to modify a strand of DNA. The first is through the process of mutations. In mutations, this means that some of the values in the string will be randomly changed. The second way to change the string of DNA is through the process of crossover. In a crossover, 2 strands of DNA will be swapped with each other. [3] Both of these methods will help provide genetic diversity and give the agent a change to be effective in the game. We found genetic algorithms to be a great way to solve this problem because it is already being used in applications for machine learning, image processing, routing problems, and optimization problems. [1]

## 3 Details

### 3.1 Problem Description

This would be an interesting project to work on as it would be a fun way to beat a game that was a large part of society at one point. The problem that we will be tackling is finding a way to keep the bird alive for an extended period of time without hitting obstacles. Additionally, this app was known for being difficult to get a high score on, so it would be an engaging task to create an algorithm that can score well.

### 3.2 Approach

In order to solve this problem, we will be using a genetic algorithm. There will be a few things that we will need to take into consideration when building this algorithm. The most important one is to make sure that the agent will be positioned to be in the middle of the two pipes when it goes between them. This means that the agent will need to either

tap if it is too low or not tap if it is too high. Additionally, we need to sure that the agent doesn't hit the ground or the sky. Sensor data such as the distance vectors to the next low and high pipe as well as the distance to ground and the sky will need to be given to the agent. The agent's choice of when to jump will be based on weights decided by the genetic algorithm of the given sensor data, much like the algorithm described in "Artificial Intelligence: A Modern Approach"[4]

### 3.3 Software

The software that we will be using for this project includes an open-source repository that already has a working Flappy-Bird game. This repository builds the game in python using the pygame library. For coding a genetic algorithm, we will also be using python.

## 4 Preliminary Work

We haven't done any preliminary work on this project ourselves. As mentioned above, we will be using an open-source repository that already has a working FlappyBird game allowing us to focus on the genetic algorithm. If we were to use a mobile game version or a precompiled version we would have to spend a nontrivial amount of time figuring out effective ways to interact with the game, such as having to perform image processing to determine the position of the bird and the pipes.

## 5 Evaluation

Our evaluation is quite simple for this game, since the only thing that we are trying to do is keep the bird alive for as long as possible. We oftentimes can just use the score as a good evaluation metric for how well the agent is doing. And since we are using a genetic algorithm, we can just use the score as a fitness function. [4] In this variation of the game it may be possible that the agent may not ever finish the game and will perpetually keep playing since difficulty doesn't scale. In this case, we will just put a time limit on the game and if the agent reaches it then we will just end the game.

## 6 Time Frame

Since we don't need to worry about building the game itself, we can focus on the genetic algorithm. Our first step is to choose the specific genetic algorithm that we will be using. We will then need to implement the genetic algorithm and test it. Lastly, we will need to complete the final report by the end of the semester.

## References

- [1] Tanweer Alam, Shamimul Qamar, Amit Dixit, and Mohamed Benaïda. 2021. Genetic algorithm: Reviews, implementations, and applications. International Association of Online Engineering, 20 pages.
- [2] Burak Kanber Lee Jacobson. 2015. *Genetic Algorithms in Java Basics* (1st. ed.). New York, NY: Apress, New York, NY.
- [3] Jose M. Moyano Sebastian Ventura, Jose M. Luna. 2022. *Genetic Algorithms*. IntechOpen.
- [4] Ming-Wei Chang Stuart J Russell, Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach* (fourth edition ed.). Hoboken, NJ: Pearson.