FLAPPY BIRD

THE BIRTH OF A GAME

A presentation by Ignacio Casado & Isaías Bueno

TABLE OF CONTENTS

01

Intro

02

Building a Game

03

Learning to Fly

04

Next Steps

01

INTRODUCTION

INTRODUCTION

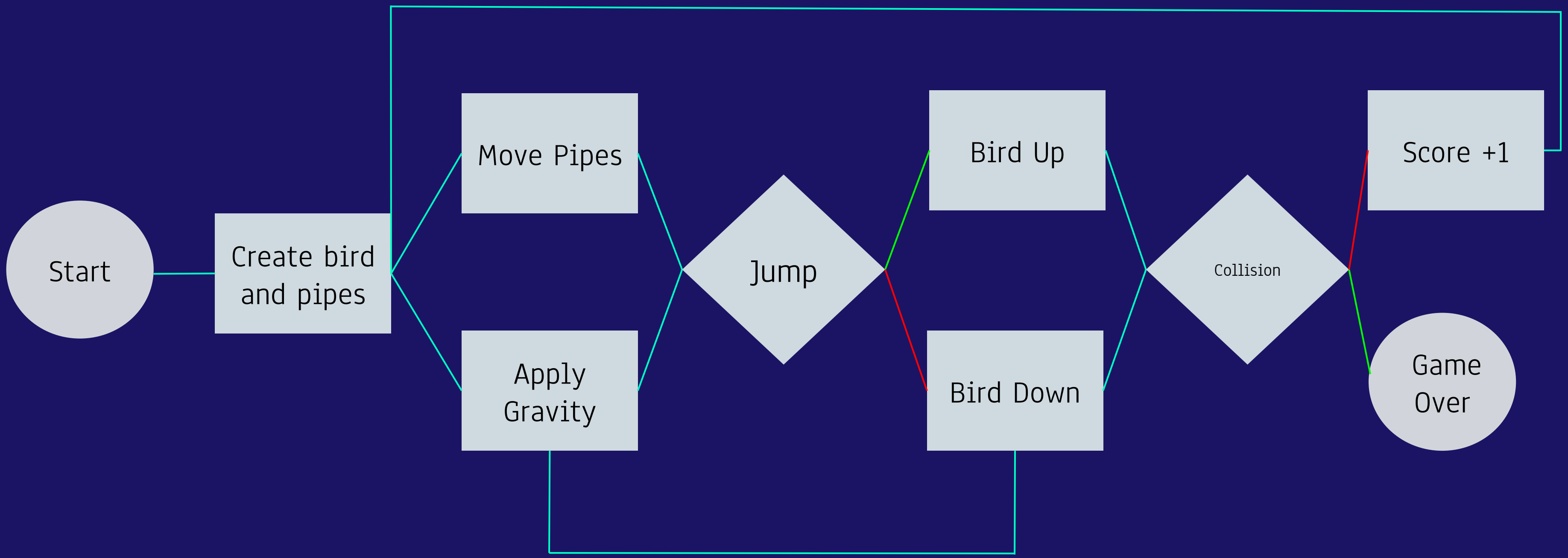
- Objectives: simulate a game in Python and make an AI play autonomously.

02

BUILDING A GAME

> “Antes de empezar a marearos con el código e ir a lo loco. Hacedme un diagrama de la dinámica del juego, por favor.”
>
> – PEPE MANZANO

FLOW CHART



Start
Create bird and pipes
Move Pipes
Apply Gravity
Jump
Bird Up
Bird Down
Collision
Score +1
Game Over

DIVING INTO THE CODE

1. Import needed libraries such as pygame.
2. Initialize pygame and set screensize
3. Define helper functions:
   a. pipeRandomHeight()
   b. gameOver()
4. Main function (loop):
   a. Set variables: player X (set) and Y (changes), gravity, jump, pipe X (set) and Y (random)
   b. Is KEYBOARD pressed? If so, jump
   c. Move pipes to the left, apply gravity and generate new pipes.
   d. Generate spontaneous items such as cherries (power-ups) and ghosts (game over)
   e. Display images on the screen
   f. Did the bird collide with any pipe, top or bottom of the screen? If so, game over

OUR CONTRIBUTIONS

1. Score on the screen
2. Game Over message appears
3. You die if you touch the ceiling or the floor.
4. Cherries with power-ups (extra score)
5. Ghosts that kill you

03

LEARNING TO FLY

BASIC CONCEPTS

Speed

Variable affecting bird movement.

Pipe_height

Variable that includes the position of both of the pipes

Player_pos

Variable reflecting player position

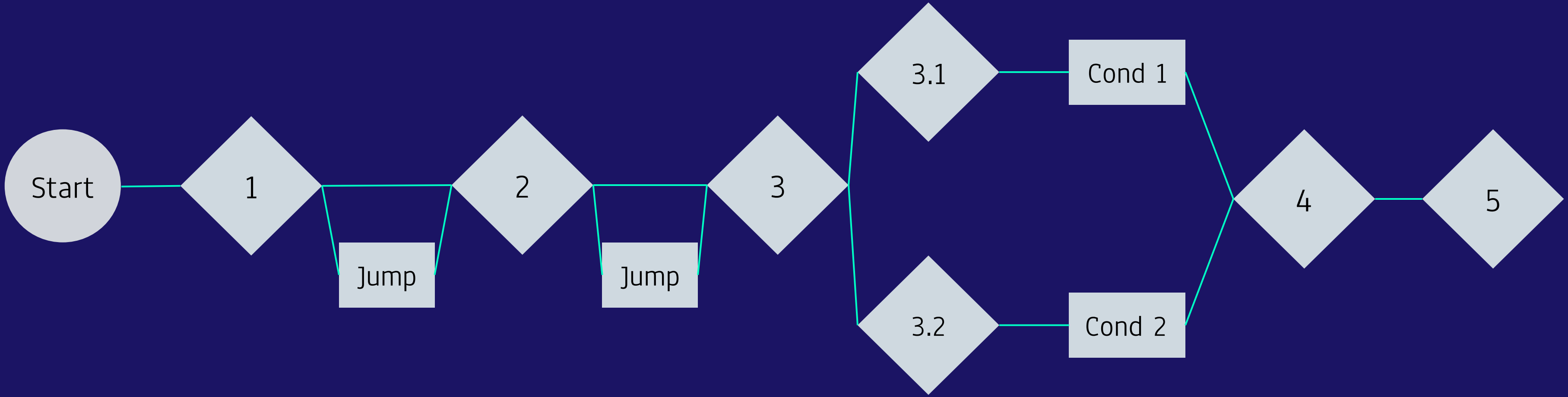
Phantom_pos

Variable reflecting ghost position

Pipe_pos

Variable including the position of the pipes on the horizontal axis

FLOW CHART AUTO-GAME



1. Check player_position > 600
2. Check speed > 13
3. Check pipe_height [1] -> [0]
4. Phantom
5. Cherry

Flappy Bird AI Gameplay

04

NEXT STEPS

NEXT STEPS

Difficulty levels

Pipes move faster or there are more obstacles

More power-ups

Extra lives or more cherries with higher point value

Start Screen

Shows game options such as difficulty

Leaderboard

Players can see their score and compare it to other player scores

Reinforcement Learning

The agent (AI) receives feedback in the form of rewards or punishments for each action it takes in the environment, and it uses this feedback to learn how to maximize its total reward over time.

A GLANCE AT REINFORCEMENT LEARNING

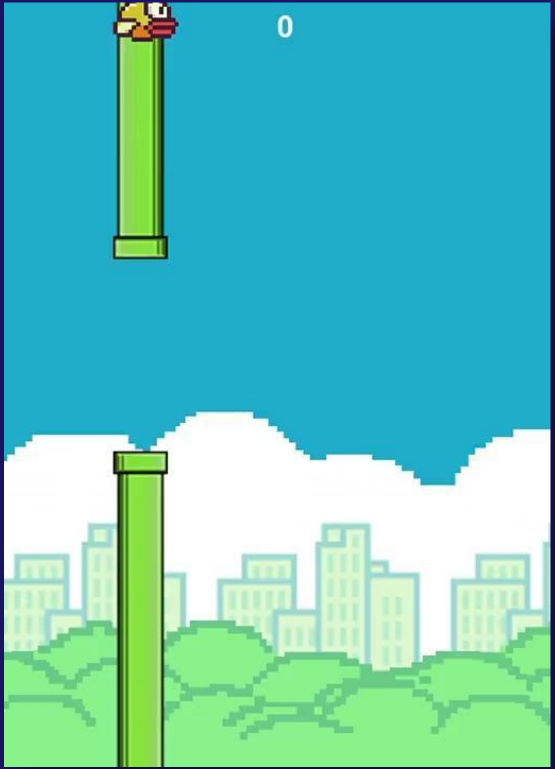
MAIN PLAYERS

FlappyBirdEnv

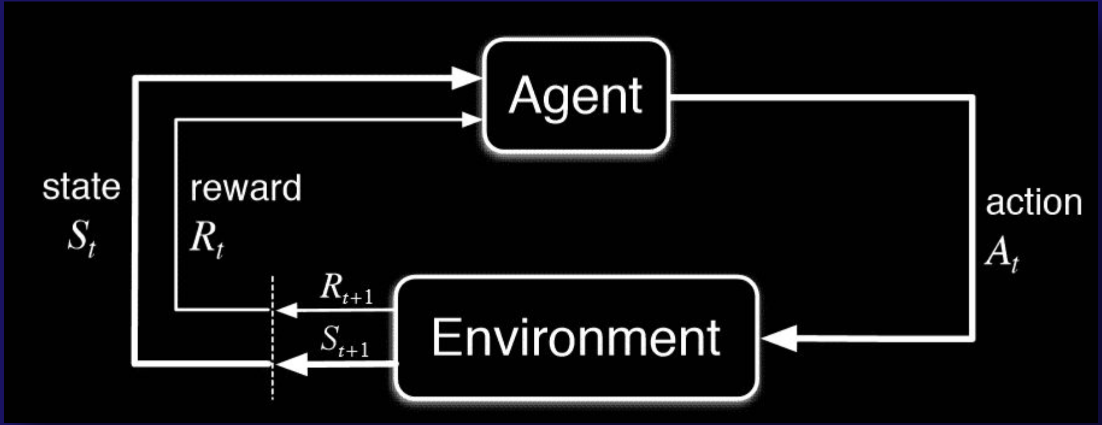
Environment of the game, including **bird and pipes position** and the **height of the two pipes**, as well as parameters such as **gravity**.

QLearningAgent

Methods to **build the Q-table**, **choose the action**, **train the model**, perform **supervised learning** and **save/load the Q-table**.



Reinforcement Learning Parameters



Learning Rate (old vs. new)
Discount Factor (now vs. later)

Exploration Rate (explore vs. exploit)
Exploration Decay Rate

THANKS!

Do you have any questions?