README

[Bird Game 3]
John Cheng

Video Link:
https://drive.google.com/file/d/1vemj3Ih3Kn37z9WNBHzR3aJzZBseDm4P/view?usp=sharing

Instructions

1. Install Python (if you don't have it already)
   Download python 3 from python.org

2. Install Pygame
   Open terminal and use command prompt *python3 -m pip install -U pygame --user*

3. Unzip bird_game3.zip file
   Open birdgame_3.py

4. Move to top of screen and press "Run" then run module

Features

1. Complex Algorithm
2. Structures keeping track of important data
3. An algorithm that works with pointing and shooting the bow and arrow keys to move the bow
4. Using randomness to decide whether to change the wind direction

Lists & Script Variables
The two main lists are arrows list and birds list, which store active arrows and the variety of birds the game offers. I also use a leaderboard list that stores dictionaries with player names and scores. Script variables include "state," "score," "wind," "game_duration," "start_time," and "last_spawn_time" to control time and activity. Moreover, each bird and arrow has its own script variables to manage movement and rendering.

Function Table

Block / Function Name	**Domain (inputs)**	**Range (outputs)**	**Behavior (role in the context of the project)**
choose_bird_type(score)	score (0 or greater)	“pigeon,” “hummingbird,” and “eagle”	Uses the player's current score plus random to decide what kind of bird spawns next. Higher score means more birds, and different bird spawns.
compute_spawn_interval(score)	score (0 or greater)	A number of milliseconds between spawns	Starts with a base delay then subtracts an amount based on the player's score. This makes birds spawn more frequently as the score increases.
maybe_change_wind(current wind, elapsed time)	A number in {-1, 0, 1} A number of milliseconds since the round started (0 or greater)	A number in {-1, 0, 1} (this is the new wind value)	Around every 2 seconds, it uses randomness to see which direction the wind changes. The wind results in how much arrows and birds are pushed sideways.

main():
update_arrows(arrows list)
	A list of arrows with position and velocity values	A new list of arrows	Moves every arrow according to velocity and removes any that have gone off-screen.

rect(self):	Arrow with position (x,y)	Bird with position (x, y)	Checks whether the arrow hits the bird’s hitbox. Check if it overlaps the hitbox too.
draw_button (rect, text):	Rectangle: the position and size of a button	A text string to show the text	Draw a button for all menu buttons.
update_leaderboard (name, score):	name: a string text	score: a non negative number	Inserts a {name, score} into a dictionary, a leaderboard list and sorts it from highest to lowest.
move(self, keys): (keys pressed)	A set of booleans	none	Moves the bow right or left depending on which

	representing whether right or left keys are pressed		keys are pressed.
rotate_aim(self): (mouse position)	Current (x,y) position of mouse	none	It shows the angle of the bow depending on the mouse’s direction. It uses trig and rotates the bow so it always points at the cursor.
load_leaderboard():	None	A list of entries, where each entry is a dictionary with name and score	Checks if the leaderboard file exists, otherwise returns an empty list.
save_leaderboard():	A list of leaderboard entries (name, score)	none	The highscores are saved between game runs.

Bug Writeup

One of the main bugs I ran into was the direction my bird sprite was facing. Whenever a bird spawned on the left and moved to the right of my screen, the sprite was facing left, and vice versa. I could reproduce this by starting the game, waiting for a bird to spawn from each side, and watching that the sprite looked backwards, relative to its motion. The root cause was my image flipping mechanic. My base sprite was drawn facing its left, but in my draw function I treated the base image as if it were facing right and flipped the image as if it were left, so I was using the wrong code for the direction. I debugged it by experimenting with pygame.transform.flip and swapping the direction values from direction == 1 to direction == -1 until the bird finally looked like it was facing the right direction. I did not make an Ed post about this particular issue, but the bug was related to the sprite orientation class and image transforms in my code.