

Trabajo Práctico Final — Programación Computacional

Flappy Fish: Juego basado en Pygame

Julieta Zanoni, Mariia Osipova, Santino Scofano y Morena
Roldan

Universidad de San Andrés

jzanoni@udesa.edu.ar

mosipova@udesa.edu.ar

sscofano@udesa.edu.ar

mroldan@udesa.edu.ar

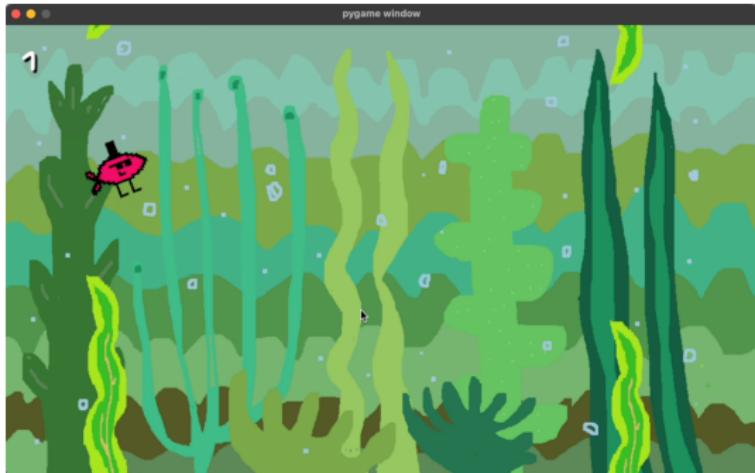
12 de diciembre 2025

Resumen de la presentación

- 1 Introducción
- 2 Arquitectura del Juego
Módulo game.py

Introducción

Nuestro trabajo práctico está dividido en dos partes: trata sobre el desarrollo de un videojuego maunal inspirado en Flappy Bird, llamado Flappy Fish, implementado en Python utilizando la librería Pygame.



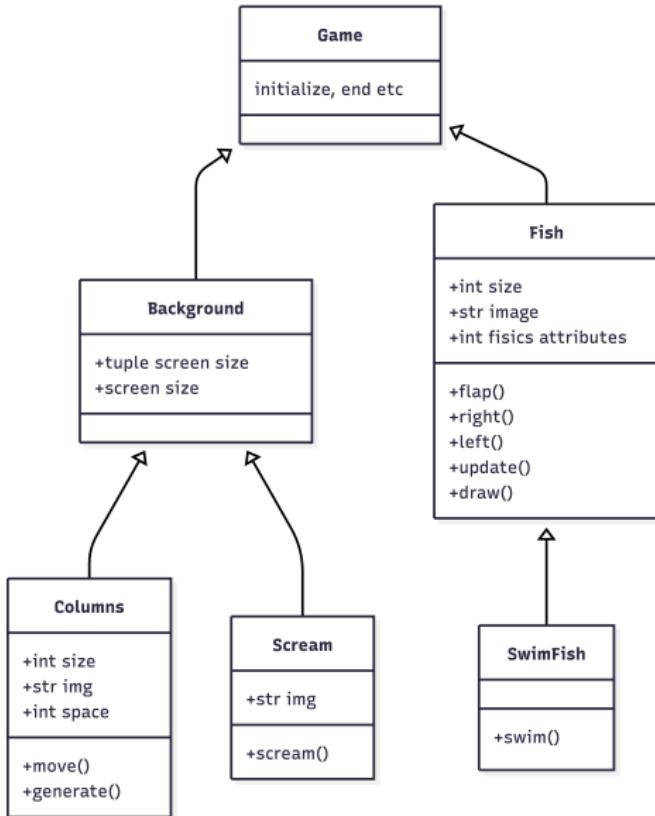
Introducción

La segunda parte del Trabajo Práctico se enfoca en la implementación de un Algoritmo Genético (AG) para entrenar a una población de “peces” a jugar de forma autónoma al videojuego.



Arquitectura del Juego

Pensando en la arquitectura del juego, nos enfrentamos al primer desafío: ¿cómo debíamos estructurar y organizar el proyecto? Comenzamos trabajando a partir de este borrador inicial.



Arquitectura del Juego

Para entender mejor cómo estructurar el proyecto, analizamos varios juegos desarrollados con Pygame y publicados de forma abierta. Estas referencias nos permitieron observar enfoques comunes de arquitectura y organización del código. Entre ellos, miramos proyectos como [Super Mario Python](#) y [Tower Defence Game](#), que utilizamos como guía conceptual.

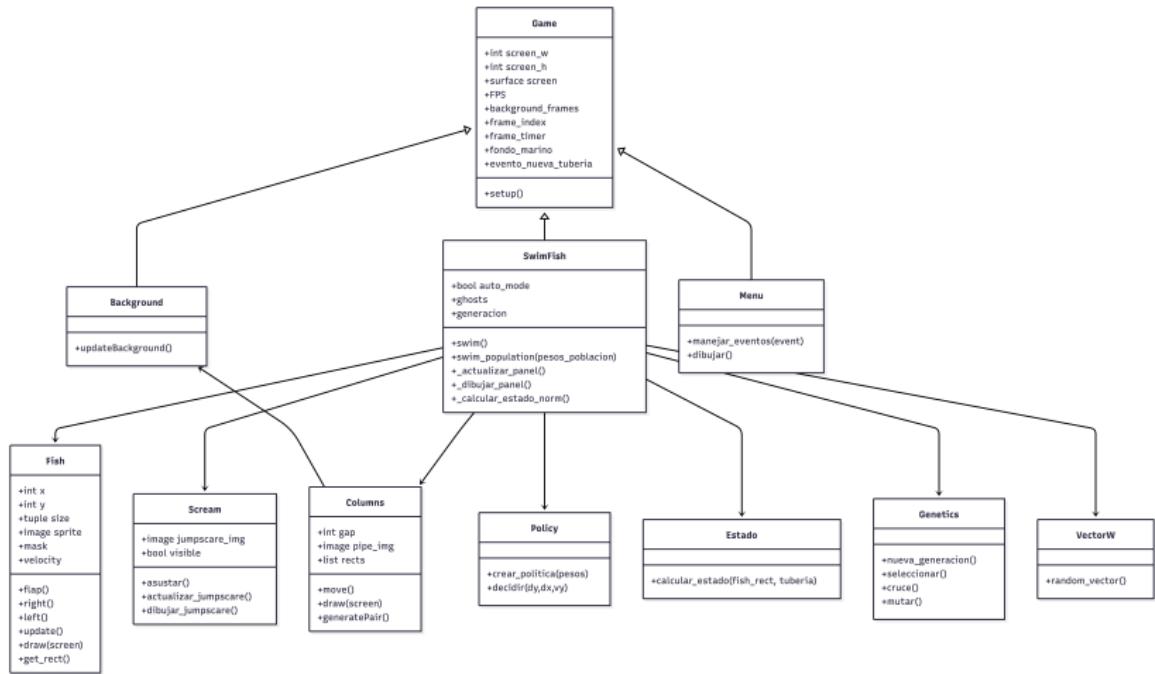
This screenshot shows the GitHub repository 'super-mario-python'. It displays a list of 266 commits from 52 contributors over 4 years. The commits are organized by file, with a summary of changes and the date of each commit. The repository has 10 branches and 0 tags. The commit history includes changes like reverting f-strings back to format(), adding attributes to entity classes, and fixing bugs related to coin rendering and jumping mechanics.

File	Summary	Date
classes	reverted f-Strings back to format()	5 years ago
entities	Added attributes to entity class to make dealing with Koo...	4 years ago
img	Revert *fix: Render coin animation with transparent backg...	5 years ago
levels	Allow mushroom to come out of box	5 years ago
sfx	added sfx when big mario is damaged and fixed 2 bugs	5 years ago
sprites	Update RedMushroom.json	5 years ago
trails	Fixed jumping in air bug	4 years ago
.gitignore	Add more files to .gitignore.	5 years ago
README.md	Update README.md	5 years ago
compile.py	Fix Formatting	5 years ago
main.py	reverted f-Strings back to format()	5 years ago
requirements.txt	Create requirements.txt	5 years ago

This screenshot shows the GitHub repository 'Tower-Defence-Game'. It displays a list of 8 commits from 1 contributor over 6 years. The commits are organized by file, with a summary of changes and the date of each commit. The repository has 1 branch and 0 tags. The commit history includes adding files via upload for enemies, main_menu, menu, and towers, as well as creating .gitpod.dockerfile and .gitpod.yml files.

File	Summary	Date
enemies	Add files via upload	6 years ago
main_menu	Add files via upload	6 years ago
menu	Add files via upload	6 years ago
towers	Add files via upload	6 years ago
.gitpod.dockerfile	Create .gitpod.dockerfile	6 years ago
.gitpod.yml	Create .gitpod.yml	6 years ago
README.md	Updated README via script	last year
game.py	Add files via upload	6 years ago
requirements.txt	Create requirements.txt	6 years ago
run.py	Add files via upload	6 years ago

Arquitectura del Juego



Arquitectura del Juego

El proyecto se estructura en los siguientes módulos:

- **game.py** — configuración general del juego.
- **fish.py** — física, movimiento y máscara del pez.
- **generacion_de_tuberias.py** — creación y movimiento de tuberías.
- **menu.py** — interfaz de menú.
- **swim_fish.py** — lógica manual y modo de Algoritmo Genético.
- **ml/** — política del agente, estado, pesos y genética.

Módulo game.py

El módulo **game.py** es el punto de inicio del sistema y es responsable de la inicialización de Pygame, así como de la configuración de todos los parámetros del entorno. En este módulo se definen constantes fundamentales como las dimensiones de la ventana, FPS y las rutas de imágenes y sonidos. Además, aquí se cargan tanto el fondo estático como la animación utilizada en el menú.