Creación de un videojuego con Godot 4

Ciclos Formativo de Grado Superior
Desarrollo de Aplicaciones Multiplataforma

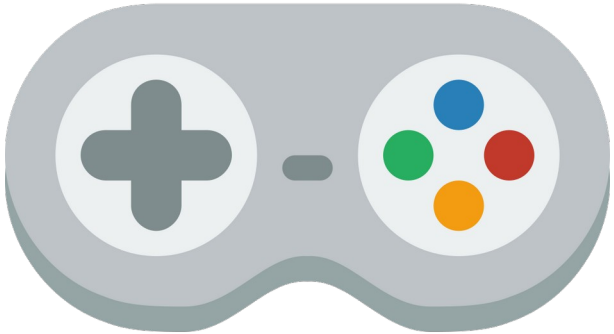
Alumno: Hugo del Rey Holgueras
Tutor: Rodrigo Iglesias Gorron

Índice

1. Motivación
2. Objetivos
3. Herramientas
4. Planificación
5. Demostración del juego
6. Posibles mejoras
7. Problemas
8. Conclusiones

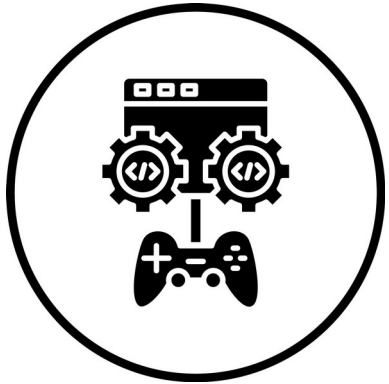
1. Motivación

- **Gusto por los videojuegos**
- **Curiosidad del motor emergente Godot**



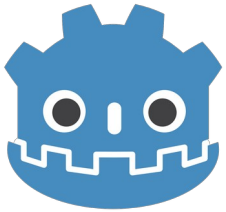
2. Objetivos

- Aplicar los conocimientos obtenidos durante el curso
- Aprender a usar un motor de videojuegos
- Aprender a organizar proyectos complejos



3. Herramientas

Desarrollo:



Godot

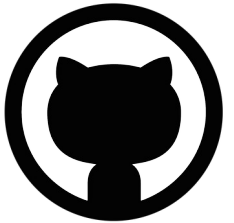
Aseprite

Base de Datos:

Supabase

Control de Versiones:

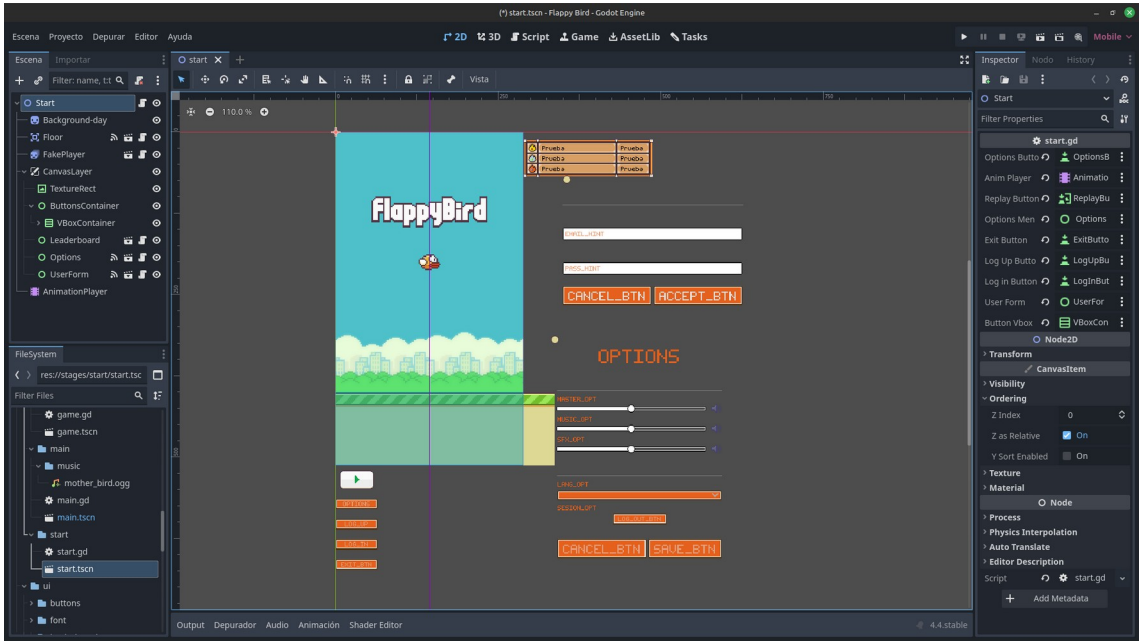
Git



GitHub

3. Herramientas

- **Godot**
  - Motor de videojuegos
  - Código abierto
  - Sin pagos ni licencias de uso
  - GDScript, su lenguaje propio



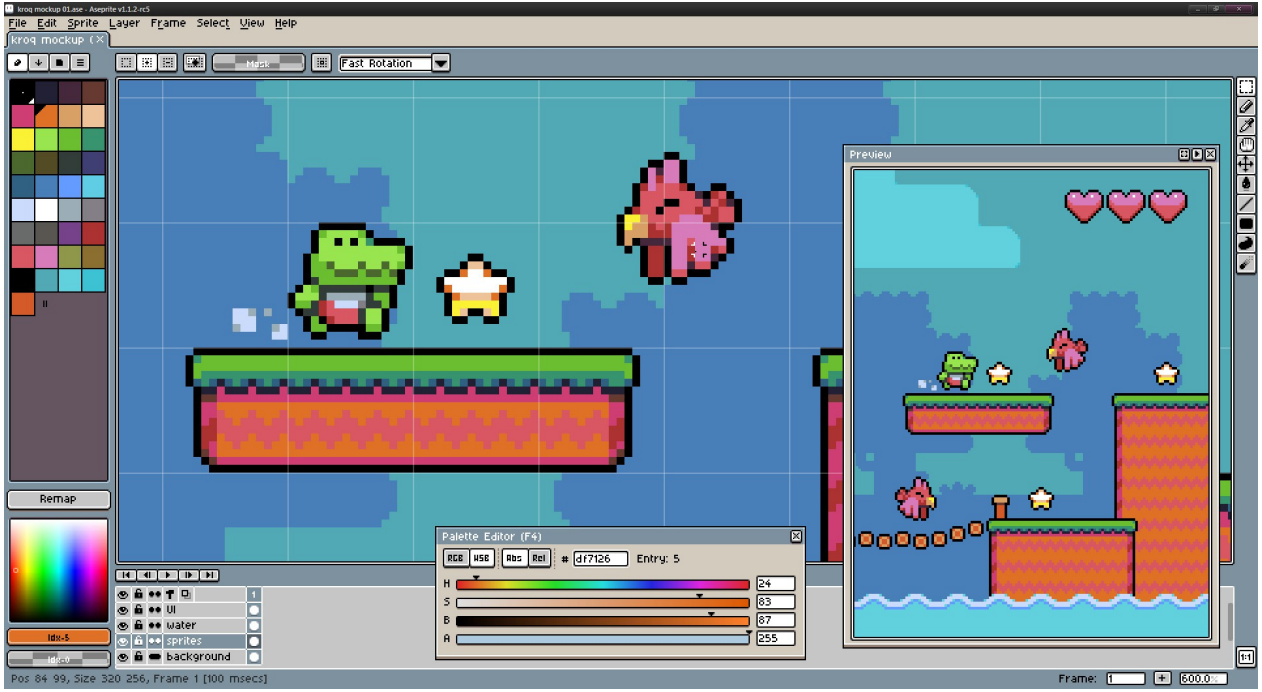
3. Herramientas

- **GDScript**
  - Sintaxis similar a Python
  - Interpretado en debug y compilado al exportar
  - Permite cambios en tiempo de ejecución

```
6   const GAME_SCENE: String = "res://stages/game/game.tscn"
7
8   @export var options_button: Button
9   @export var anim_player: AnimationPlayer
10  @export var replay_button: TextureButton
11  @export var options_menu: OptionMenu
12  @export var exit_button: Button
13  @export var log_up_button: Button
14  @export var log_in_button: Button
15  @export var user_form: UserForm
16  @export var button_vbox: VBoxContainer
17
18
19  # Ajustes iniciales de start
20  func _ready() -> void:
21      Supabase.auth.signed_in.connect(_on_log)
22      Supabase.auth.signed_up.connect(_on_log)
23      Supabase.auth.token_refreshed.connect(_on_log)
24      Supabase.auth.signed_out.connect(_on_log_out)
25      EventBus.locale_changed.connect(_on_locale_changed)
26
27      if ConfigSaveHandler.loading_state == ConfigSaveHandler.LoadingState.LOADED:
28          anim_player.play("show_buttons_auto")
29      else:
30          ConfigSaveHandler.config_ended.connect(_on_config_ended)
31
32      _hide_log_buttons(Supabase.auth.client != null)
```

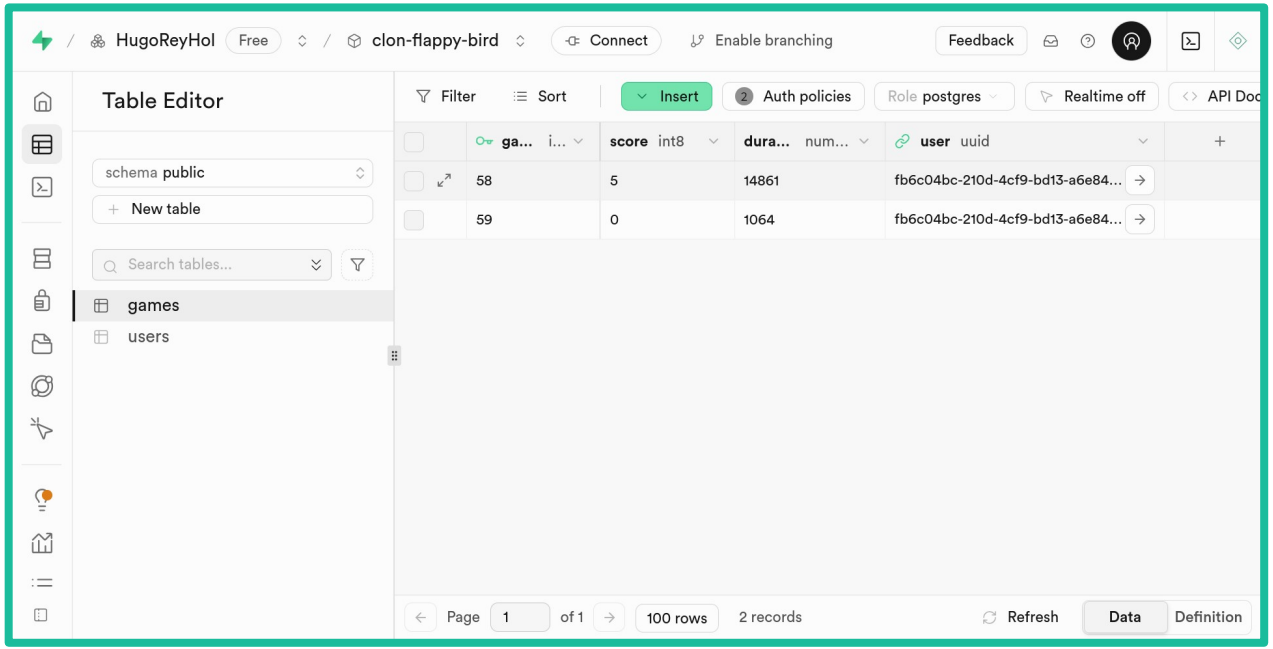
3. Herramientas

- **Aseprite**
  - Editor de imágenes pixel art



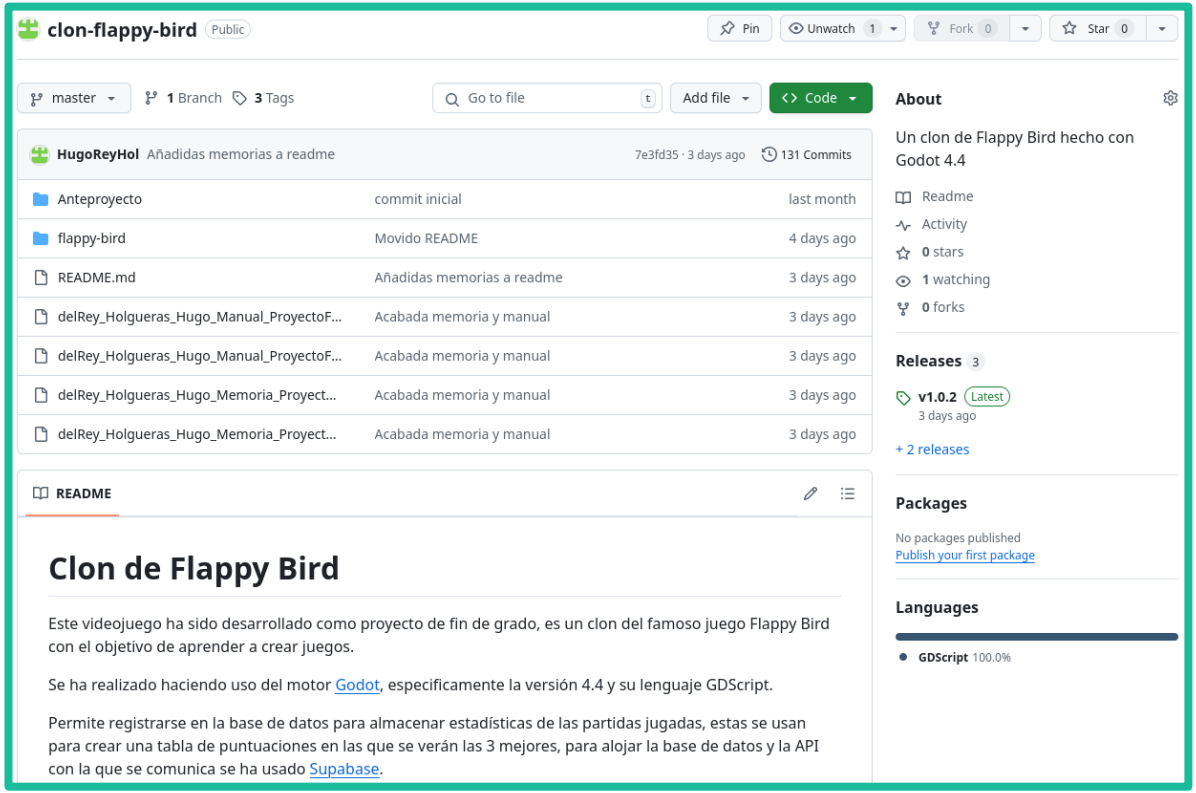
3. Herramientas

- **Supabase**
  - Base de datos PostgreSQL
  - API REST generada automáticamente
  - Plan básico gratuito



3. Herramientas

- **Git**
  - Control de versiones
- **Github**
  - Repositorios remotos
  - Permite publicar lanzamientos



4. Planificación

- **GDD (Game Design Document)**

✔ ***Realizados***

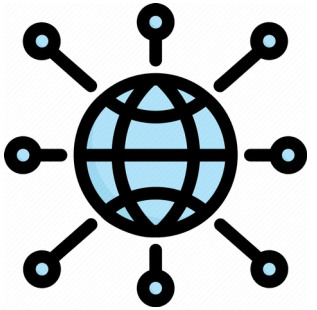
- Información
- Mecánicas
- Arte y Audio
- Tecnologías
- Interfaz de Usuario

✗ ***No Realizados***

- Mundo
- Historia y Narrativa
- Monetización
- Producción y Distribución

4. Planificación

- ***No Realizados***
  - **Mundo:** Ambientación, Facciones, Zonas...
  - **Historia y Narrativa:** Sinopsis, Misiones, Diálogos...
  - **Monetización:** Modelo de Negocio, Publicidad, Microtransacciones...
  - **Producción y Distribución:** Cronograma, Responsables, Presupuesto...



4. Planificación

- ***Realizados***
  - **Información:** Detalles generales

- **Título:** Clon de Flappy Bird
- **Género:** Acción y Scroll Infinito
- **Plataformas:** Windows, Linux y Android
- **Público objetivo:** Jugadores de todas las edades
- **Descripción:** Es un juego en el que los jugadores tendrán que poner a prueba su habilidad para esquivar obstáculos para alcanzar una mayor puntuación usando la mecánica principal del salto

4. Planificación

- ***Realizados***
  - **Mecánicas:** Interacciones del usuario con el juego

En este apartado se define como interactuará el jugador con nuestro videojuego.

- **Gameplay Loop:** El jugador evitará chocarse con obstáculos que aparecerán de forma aleatoria, al colisionar con uno o salirse del mapa perderán la partida y empezarán de nuevo.
- **Controles y Cámara:** Al jugar en PC podremos saltar usando la tecla de espacio o con un clic izquierdo, mientras que en móviles pulsaremos la pantalla, en cuanto a la cámara esta estará fija en la misma posición junto con el personaje del jugador y lo único que se desplazará serán los obstáculos.

4. Planificación

- ***Realizados***
  - **Arte y Audio:** Estilo, efectos, animaciones...

> En el juego original el estilo visual es tipo *pixel art*[1], un estilo artístico que consiste en representar imágenes usando resoluciones muy pequeñas imitando la estética de los videojuegos de los años 80 y 90.
>
> El apartado sonoro se compone de sonidos de corta duración y muy distinguibles entre sí que ayudan al jugador a entender que está pasando en su partida.

4. Planificación

- ***Realizados***
  - **Tecnologías:** Las herramientas del apartado anterior

Motor: Usaré Godot por su curva de aprendizaje accesible y no tener ningún tipo de coste para el desarrollador.

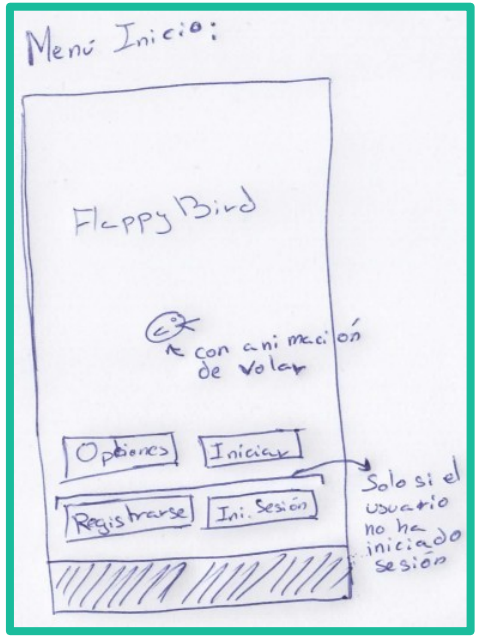
Lenguaje de Programación: Se usará GDScript ya que Godot está diseñado para usar este.

Plataforma de Desarrollo: Todo desarrollo y las pruebas unitarias se realizarán en un PC con Linux.

4. Planificación

- ***Realizados***
  - **Interfaz de Usuario:** Menús y elementos en pantalla

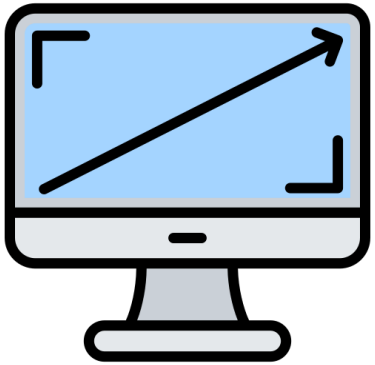
Este es el menú que aparece cuando se abre el juego, esta se compone por el título con el nombre del juego en la parte superior, en la parte central aparece el personaje con el que va a jugar el usuario y en la parte inferior hay cuatro botones, el primero permite abrir el menú de opciones, el segundo cambia a la pantalla **Partida**, y los dos últimos permiten registrarse e iniciar sesión respectivamente y solo se mostrarán en caso de que el usuario no haya iniciado sesión.



5. Demostración

6. Posibles mejoras

- Más traducciones
- Menú para ver puntuaciones
- Más variedad de obstáculos
- Mejorar el re-escalado



7. Problemas

- **Falta de documentación de herramientas**
  - Godot
  - Plugins

7. Problemas

```
68    # Allow your users to sign up and create a new account.
69    func sign_up(email : String, password : String) -> AuthTask:
70        if _auth != "": return _check_auth()
71        var payload : Dictionary = {"email":email, "password":password}
72        var auth_task : AuthTask = AuthTask.new()._setup(
73            AuthTask.Task.SIGNUP,
74            _config.supabaseUrl + _signup_endpoint,
75            _header,
76            JSON.stringify(payload)
77        )
78        _process_task(auth_task)
79        return auth_task
```

7. Problemas

```
26  func match_code(code: int = Task.NONE) -> int:
27      match code:
28          Task.SIGNIN, Task.SIGNUP, Task.LOGOUT, Task.MAGICLINK, Task.RECOVER, Task.REFRESH, Task.INVITE,
29              return HTTPClient.METHOD_POST
30          Task.UPDATE:
31              return HTTPClient.METHOD_PUT
32          _, Task.USER:
33              return HTTPClient.METHOD_GET
```

7. Problemas

```
287    # Process a specific task
288    func _process_task(task : AuthTask, _fake : bool = false) -> void:
289        task.completed.connect(_on_task_completed)
290        if _fake:
291            await get_tree().create_timer(0.5).timeout
292            task.complete(task.user, task.data, task.error)
293        else:
294            var httprequest : HTTPRequest = HTTPRequest.new()
295            add_child(httprequest)
296            task.push_request(httprequest)
```

7. Problemas

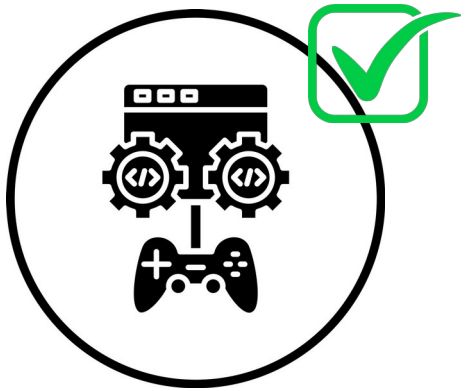
```
33  func push_request(httprequest : HTTPRequest) -> void:
34      reference()
35      httprequest.request_completed.connect(_on_task_completed.bind(httprequest))
36      httprequest.request(_endpoint, _headers, _method, _payload)
```

7. Problemas

```
299 func _on_task_completed(task : AuthTask) -> void:
300     if task.error != null:
301         error.emit(task.error)
302     else:
303         if task.user != null:
304             client = task.user
305             _auth = client.access_token
306             _expires_in = client.expires_in
307             match task._code:
308                 AuthTask.Task.SIGNUP:
309                     signed_up.emit(client)
310                 AuthTask.Task.SIGNUPPHONEPASSWORD:
311                     signed_up_phone.emit(client)
```

8. Conclusiones

- He desarrollado un juego multiplataforma funcional
- He aprendido lo básico de Godot
- He mejorado mi organización de proyectos



Fin

Contacto: hugorey@hotmail.es
GitHub: HugoReyHol

Gracias por vuestra atención

github.com/HugoReyHol/clon-flappy-bird/releases/latest