

Alberto el Pingüino

La Colonia Perdida

Índice

1. Historia

- 1.1. Personajes**
- 1.2. Marco**
- 1.3. Narrativa**

2. Gameplay

- 2.1. Game Loop**
- 2.2. Mecánicas**
- 2.3. Enemigos**
- 2.4. Diseño molecular**

3. Diseño del nivel

- 3.1. Progresión por el nivel**
- 3.2. Escenario**
- 3.3. Game Objects**
- 3.4. Escenas**
- 3.5. Scripts**

4. Arte

5. Interfaz/Controles

6. Audiencia

- 6.1. Jugadores objetivo**
- 6.2. Motivación del jugador**

1. HISTORIA

1.1 Personajes

Alberto: Alberto el pingüino es el protagonista de esta historia. Los sucesos de la historia se resuelven alrededor de nuestro héroe. Es un pingüino con habilidades nunca vistas. Ha perdido a su familia y su motivación principal es recuperarlos.

Familia: La familia de Alberto, los secuestrados. Su papel en la historia de este videojuego es ser rescatados.

Antagonistas: los personajes que tratarán de impedir que Alberto se reúna con su familia y sean felices. Tendremos unos leopardos marinos que se deslizarán por el suelo y unas orcas que nos sorprenderán desde el mar cuando Alberto trate de cruzar dos terrenos separados por mar.

1.2 Marco

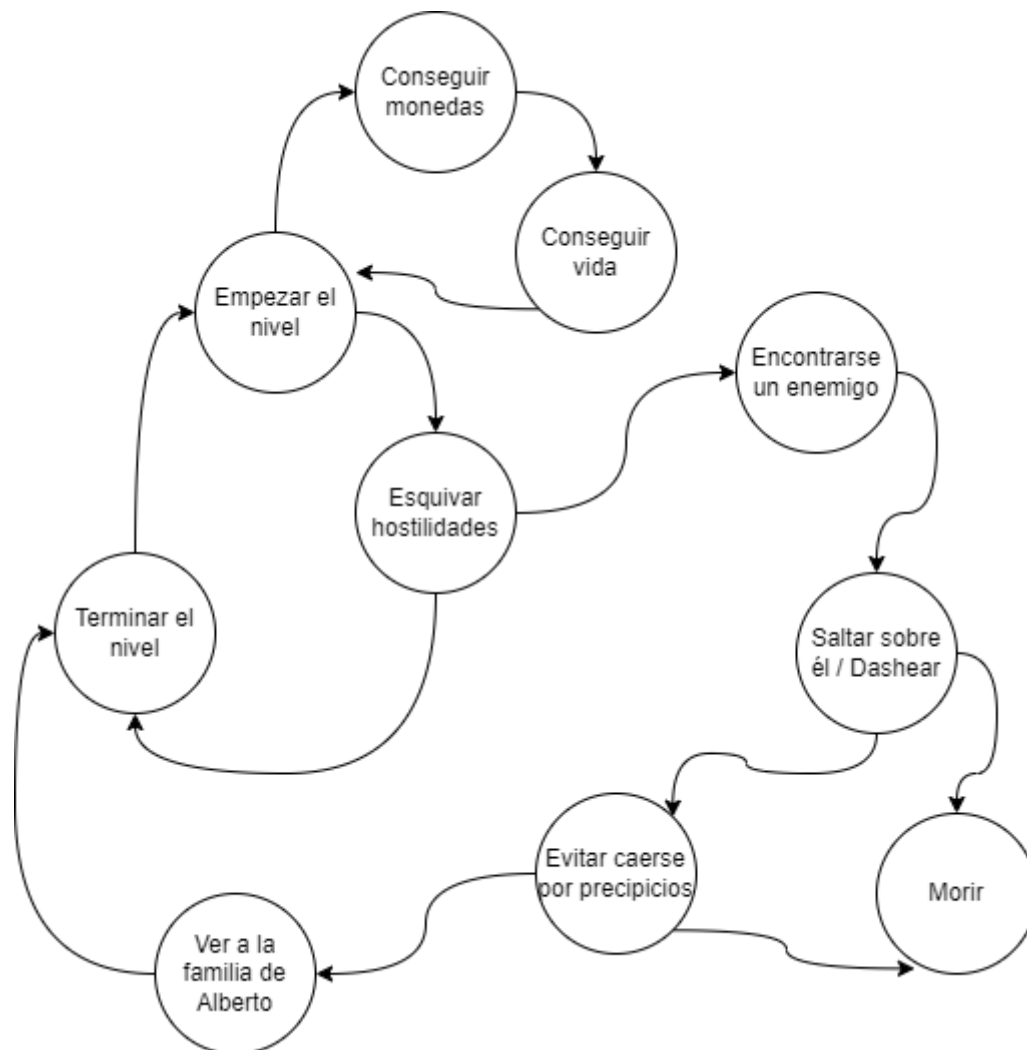
Alberto el Pingüino, La Colonia Perdida se sitúa en un gélido y cruel escenario, lleno de hostilidades para nuestro protagonista. Este paisaje estará lleno de enemigos y accidentes geográficos diversos los cuales Alberto tendrá que evitar para poder seguir adelante hacia su objetivo.

1.3 Narrativa

Viviremos en una aventura con nuestro protagonista, un pingüino llamado Alberto, quien lamentablemente ha perdido a su familia. Alberto está decidido a reunirse con ellos y, para hacerlo, deberá recorrer un mundo cubierto de nieve. A medida que Alberto avanza por el mundo, se encontrará con una serie de desafíos emocionantes y enemigos que intentarán detenerlo en su búsqueda. A lo largo de su viaje, descubrirá pistas y secretos que lo llevarán más cerca de la verdad detrás de la desaparición de su familia. ¿Tienes lo que se necesita para ayudar a Alberto a resolver el misterio, rescatar a su familia? Acompaña a nuestro valiente pingüino en esta emocionante aventura a través de un mundo nevado lleno de desafíos y sorpresas.

2. Gameplay

2.1 Game Loop



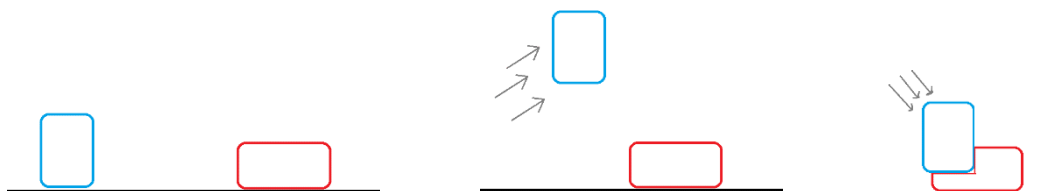
2.2 Mecánicas

La mecánica principal de este juego se basa en el de cualquier juego plataforma 2D como el Super Mario Bros. En este caso, nuestro objetivo será basarnos principalmente en el power-up de pingüino introducido en el New Super Mario Bros Wii.

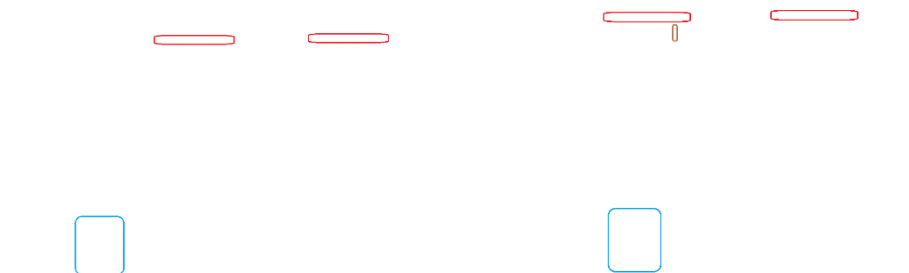


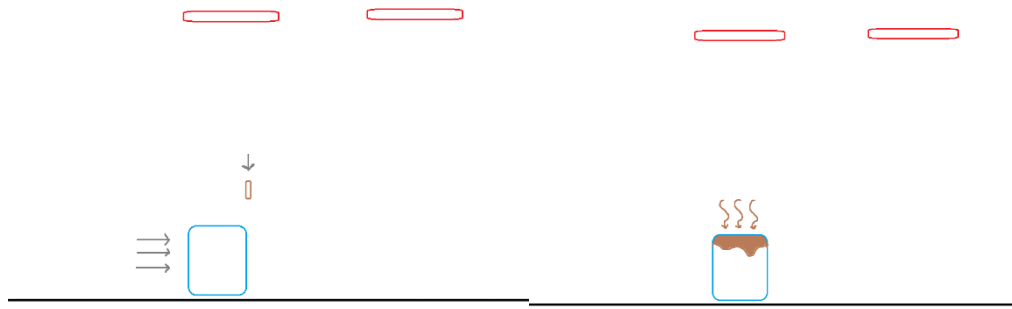
Aquí tenemos otros ejemplos de actividades realizadas en el curso de pasar un nivel:

Salto y eliminación de enemigo en la tierra

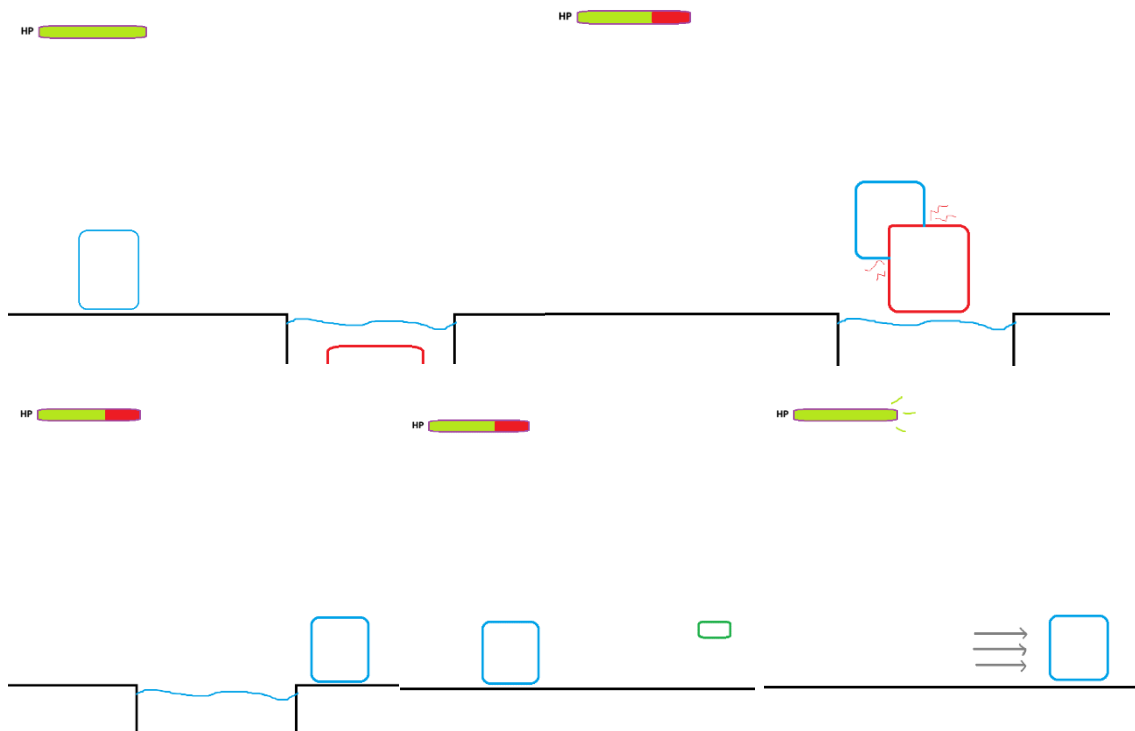


Enemigos voladores que si te da su proyectil te ralentiza





Daño recibido por un enemigo que sale del agua y cura mediante un objeto encontrado en el nivel



En el tema de la vida y las curas nos basaremos principalmente en el sistema hecho el juegos como el Kirby`s Return to Dreamland



2.3 Enemigos

En este juego nos encontraremos con 3 clases de enemigos principales:

Leopardo marino: Enemigo terrestre. Hará daño cuando te toque.

Orca: Enemigo que saldrá saltando de ciertos barrancos por el nivel. Hará daño en contacto.

Petrel: Enemigo aéreo. Se dedicará a lanzar proyectiles que ralentizarán al jugador.

2.4 Diseño molecular

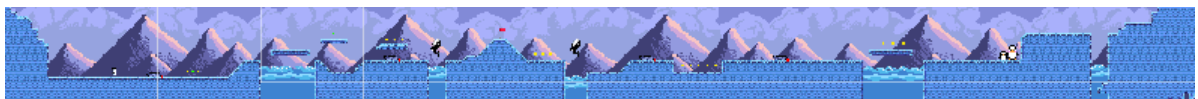
Tenemos un nivel con un diseño completamente lineal con sectores muy distinguidos.



3. Diseño del nivel

3.1 Progresión por el nivel

El nivel de prueba realizado por ahora es un prototipo de lo que tenemos ideado que sean los siguientes niveles. Es un diseño básico de plataformas en el cual tendrás que saltar para avanzar por el mismo. Le hemos introducido un sistema de “checkpoints” para poder guardar tu progreso a mitad del nivel. Al final del mismo, se encuentran los familiares, si se llega a ellos, terminará el nivel. Aquí tenemos el nivel en cuestión.



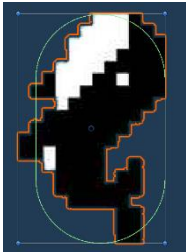
3.2 Escenario

El escenario sobre el que se sitúa nuestro videojuego es un único escenario, un terreno gélido con ciertos accidentes geográficos los cuales tendrá que esquivar Alberto el Pingüino.

3.3 Game Objects



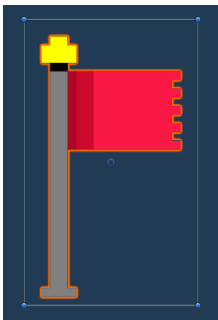
Jugador: Objeto que controlará el jugador.



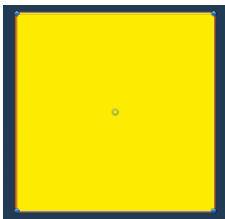
Orca: Uno de los dos enemigos del juego. Salta del agua y mata al jugador de un golpe



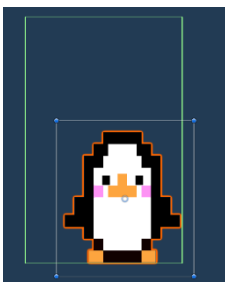
Leopardo marino: Enemigo terrestre. Va de lado a lado. Te quita una vida con la colisión y puede ser matado saltándole encima.



Checkpoint: Guarda posición para la reaparición del jugador en el evento de su muerte.



Krill/Sardinas: Objeto que proporcionará vida y puntuación al jugador. 5 Krill equivalen a 1 Sardina. Con 1 Sardina recuperas un punto de vida.



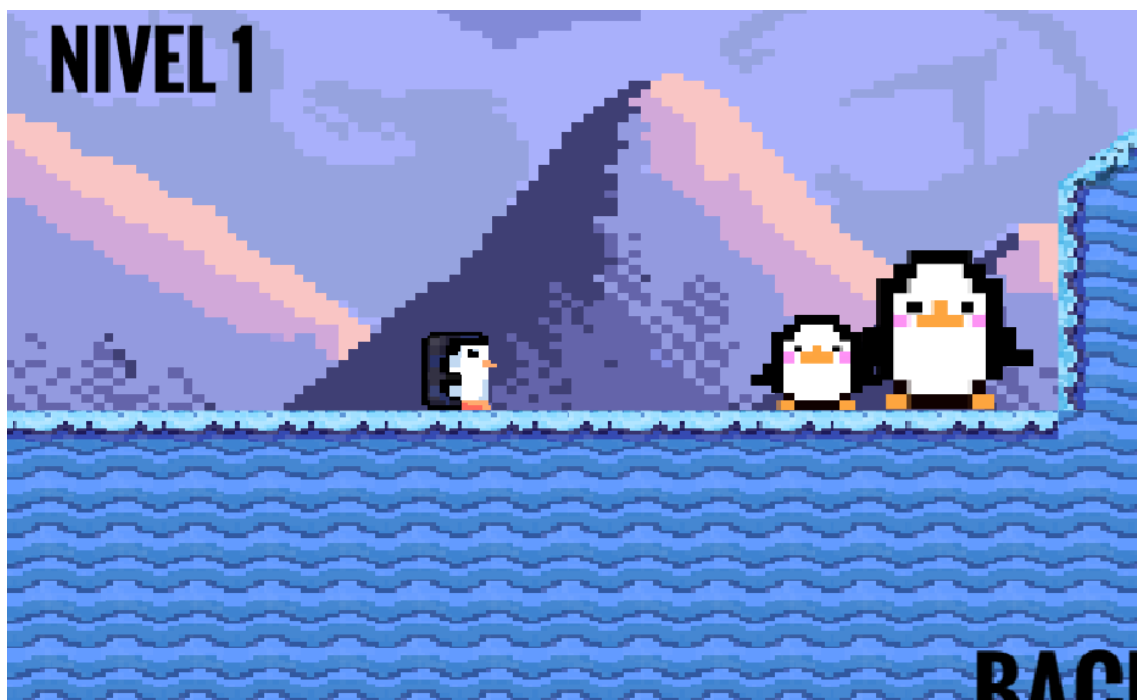
Familia/Fin de partida: Objeto con el cual colisionas y terminas el nivel. Son la familia de Alberto. Objetivo final del juego.

3.4 Escenas

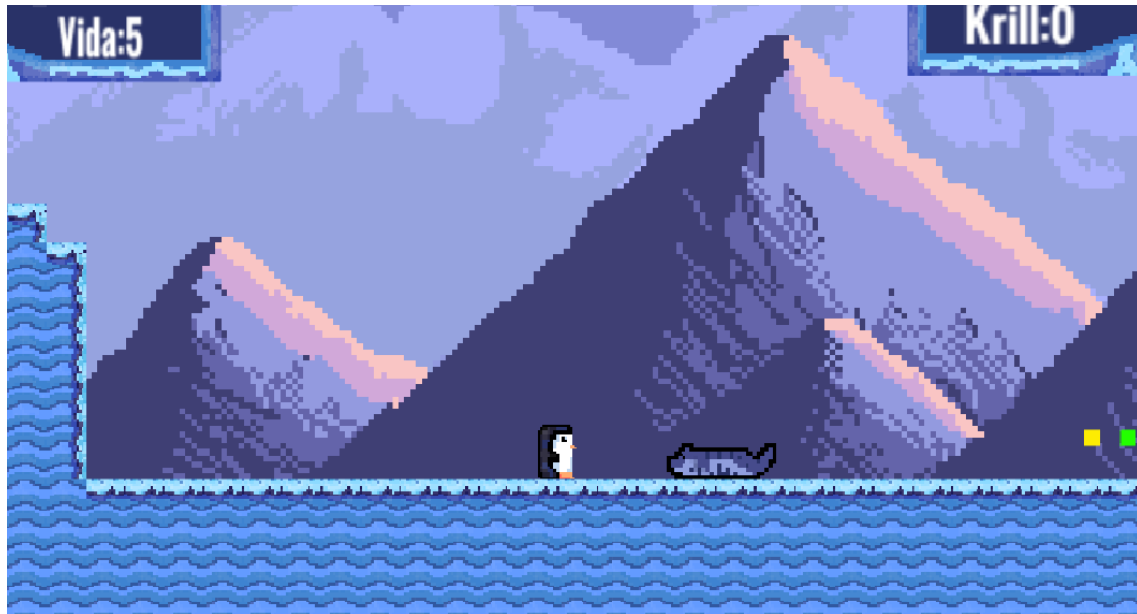
Inicio: Pantalla de inicio con el título del juego. Seleccionas o bien empezar la partida o bien salir del juego.



Selección nivel: Pantalla donde seleccionarás el nivel que quieres jugar (En nuestro caso el nivel 1).



Nivel1TileMap: El nivel en el que el jugará el jugador. Lleno de enemigos y obstáculos para dificultarle la llegada al final.



3.3 Scripts

Aquí nos encontramos los distintos scripts:



Camara.cs: Controla la cámara.

CheckpointController.cs: Controla el checkpoint.

Leopardo.cs: Controla el movimiento del leopardo marino.

LevelFinshed.cs: Controla el canvas del final de nivel.

MenuInicio.cs: Controla el menú de inicio.

MenuNiveles.cs: Controla el menú de selección de niveles.

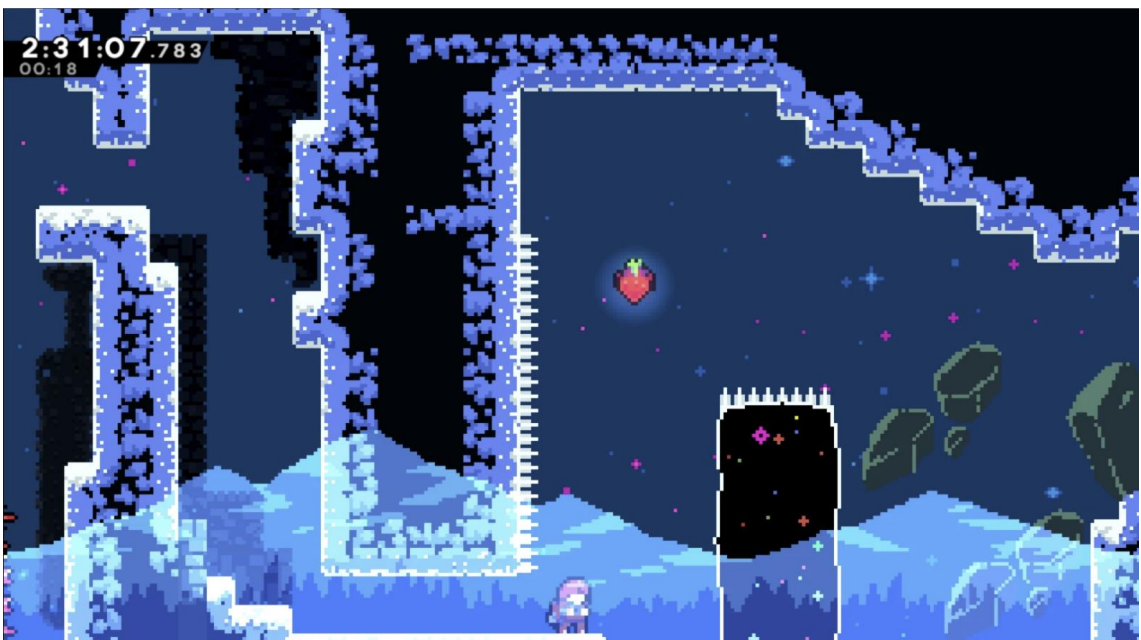
MovimientoGuino: Controla tanto el movimiento de Alberto como las colisiones con el resto del entorno.

movimientoorca.cs: Controla el movimiento de la orca de salto.

PetrelMovement.cs: No implementado.

4. Arte

El arte de este juego será basada principalmente en el “pixel art” vista en infinidad de títulos. Principalmente nos basaremos en el arte del juego llamado Celeste. Otro plataformas 2D con una gran variedad escenarios gélidos.



5. Interfaz/Controles

Este juego estará implementado únicamente para PC, por tanto, sus controles principales serán WASD para movimiento (W arriba, A izquierda, D derecha y S abajo) y espacio para saltar.

(Completar interfaz)

6. Audiencia

6.1 Jugadores objetivo

Jugadores de todas las edades: jugadores de todas las edades, desde niños pequeños hasta adultos.

Jugadores casuales: “Alberto el pingüino” es un juego con mecánicas sencillas y con poca dificultad lo convierte en una opción ideal para cualquier tipo de jugador, ya sea experimentado o casual.

Fanáticos por los juegos de plataforma

6.2 Motivación del jugador

El principal motivador en este juego es pasarse el nivel, además de intentar completarlo con la puntuación más alta, la cual se conseguirá recogiendo “coins” o eliminando enemigos.

INSTRUCCIONES SOBRE LA PLANIFICACIÓN

1. Contenido Product Backlog:

a. Macro-tareas (big issues):

- i. Diseño de las mecánicas de todos los personajes:
 - Pingüino: saltar, avanzar, retroceder, deslizarse, (ataque a distancia).
 - Orca: Saltar.
 - Petreles: volar, atacar
 - Leopardos marinos: avanzar, retroceder.
 - Krill: estático.
 - Sardina: estático o saltar.
- ii. Interfaz:
 - Marcador de impactos y número de vidas y puntos.
- iii. Diseño gráfico de los niveles:
 - Diseño de los distintos tipos de suelo y las mecánicas que los personajes adquieren al estar en ese suelo: nieve hielo (agua si avanzamos óptimamente).

iv. Diseño del sistema de puntuación y vida:

- Plantear cuánta vida va a tener Alberto dentro del nivel, es decir, cuántos impactos va a poder recibir antes de perder una vida y el daño que recibe de cada enemigo. También cuántas vidas va a tener. Queremos implementar un sistema de cura de vida para recuperar si consumes ciertos elementos en el nivel (sardinas, krill). También plantear cómo va a ir el sistema de puntuación y si recuperas una vida con una cantidad determinada de puntos.

v. Menús variados: desarrollar los distintos menús como pausa, inicio, página de muerte, etc.

b. Cronograma en escala de semanas:

Iteración 1: en esta iteración planificamos y repartimos entre los miembros del grupo las tareas, según los roles asignados para crear el concepto de nuestro juego. Estimación de 1 semana.

Iteración 2: en esta iteración empezamos a desarrollar las tareas propuestas, comprobando y revisando el funcionamiento del juego con pruebas unitarias. Estimación entre 1 y 2 semanas.

Semana del 2 de octubre:

- Storyboard digital mecánicas
- Storyboard en digital narrativa
- Documentación extensa sobre el juego
- Acordar el sistema de puntuación y de vidas
- Acordar cuantos tipos de suelo vamos a tener.

Semana del 9 de octubre:

- Mecánica del pingüino (salto, desplazamiento)
- Mecánica de la orca (salto)
- Mecánica del petrel (movimiento)
- Mecánica del leopardo marino (movimiento)
- Diseño del nivel de prueba (en papel)

Semana del 16 de octubre:

- Pasar el nivel de prueba a unity
- Interacción de los personajes con los distintos tipos de suelo.
- Ataque de los enemigos

Iteración 3: en esta iteración vemos si podemos añadir alguna funcionalidad más al juego para mejorarlo (como niveles extra o nuevas mecánicas de los personajes) y corregir posibles problemas que surgieran a la hora de implementar estas nuevas mecánicas. Estimación entre 1 y 2 semanas.

Semana del 30 de octubre:

- Empezar a hacer mecánicas de enemigos
- Refinar mecánicas jugador
- Empezar diseños básicos para personajes
- Mirar software para hacer diseños para el mapa
- Empezar a ver mecánicas de cura y sistema vida

Semana del 6 de noviembre:

- Seguir refinando mecánicas jugador
- Diseños de ítems en el mapa (Curas, krill)
- Seguir con el diseño del mapa
- Complicar movimientos de los enemigos
- Programar proyectiles lanzados por los petreles
- Implementar mapa con el diseño del terreno
- Ver posibles tipos de terreno (hielo, "one-way platforms")

Semana del 13 de noviembre:

- Repasar mecánicas jugador
- Adaptar el mapa a las nuevas mecánicas
- Repartir enemigos por el mapa
- Repartir curas y krill por el mapa

Iteración 4: en esta iteración vamos a buscar inicialmente corregir los errores que nos han sido expuestos mediante las entrevistas realizadas a las diferentes personas. Los dos principales

errores a corregir son de accesibilidad y de información proporcionada al jugador. Tenemos que darles pistas sobre cómo usar mecánicas más complejas y de para qué sirven las cosas que van sucediendo a lo largo que pasas el nivel, como por ejemplo, el checkpoint.

Semana del 27 de noviembre:

- Arreglar cosas básicas de interfaz
- Empezar con animaciones
- Arreglar lo de las pistas al usuario

Semana del 4 de diciembre:

- Seguir con animaciones
 - Empezar con enemigos y deslizamiento pingüino
- Considerar poner partículas de movimiento
- Refinar deslizamiento

Semana del 11 de diciembre:

- Hacer documentación

c. Roles y responsabilidades de cada miembro:

- Team lead (responsable de la gestión de su equipo): Gabi
- Programmers (encargados de la programación de los niveles, funcionamiento de los personajes, etc.): Miguel, Jorge
- Level designers (encargados de diseñar los niveles de juego y su interfaz): Gabriel, Álvaro, Rodrigo
- Quality assessment: (encargados de evaluar la calidad y jugabilidad del juego) Rubén

d. Lista de recursos necesarios (herramientas software): Unity, Visual Studio, Pixel Art, alguna herramienta web que necesitemos, etc.

Contenidos: [Link al Trello con las tareas y las distintas agrupaciones para las tareas.](#)

[Link al repositorio en GitHub.](#)(Contiene ejecutable del juego, pdfs con el resultado de los user tests y Excel con las tareas)

Retrospectiva

Riesgos

El mayor riesgo que nos seguimos encontrando es el de no llegar a tiempo por intentar abarcar más de lo que podemos. También corremos el riesgo de muchas veces no organizarnos bien debido a la falta de comunicación.

Problemas de retrasos

Uno de los mayores problemas que hemos encontrado a la hora de progresar en esta iteración ha sido que algunos integrantes de nuestro equipo han tenido bastantes problemas a la hora de usar las herramientas Git. Eso nos ha retrasado porque no hemos podido avanzar en ciertos ámbitos. Tampoco hemos sabido muy bien cómo organizarnos en ciertos momentos y la comunicación podría haber sido mejor.

Ayuda al equipo

Nos hemos ayudado mucho tanto de distintos sitios webs como de Youtube a la hora de manejar posibles conflictos con Github, Unity, etc. Hemos usado también algunos Assets de la tienda de Unity. También a la hora de realizar diferentes scripts nos hemos apoyado de código disponible en GitHub.

Meta

Lo que buscamos en esta última iteración es dejar un producto que sea lo mayor disfrutable posible. Hemos visto que hemos tenido bastantes fallos a la hora de plantear ciertas cosas. Lo que hemos visto reflejado principalmente a la hora de hacer BetaTesting en los entrevistados ha sido que había ciertas partes en la navegación de los menús que no eran para nada intuitivas, al igual que algunas mecánicas que teníamos propuestas para el jugador. También vamos a intentar implementar algunas animaciones para que el jugador se sienta más inmerso en el juego. En general es una meta totalmente realista, pero difícil de conseguir que salga totalmente como queremos.