

# **Alberto el Pingüino**

## **La Colonia Perdida**

### **Índice**

#### **1. Historia**

**1.1. Personajes**

**1.2. Marco**

**1.3. Narrativa**

#### **2. Gameplay**

**2.1. Core Loop**

**2.2. Mecánicas**

**2.3. Enemigos**

#### **3. Diseño del nivel**

**3.1. Progresión por el nivel**

**3.2. Escenario**

#### **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.

## **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.

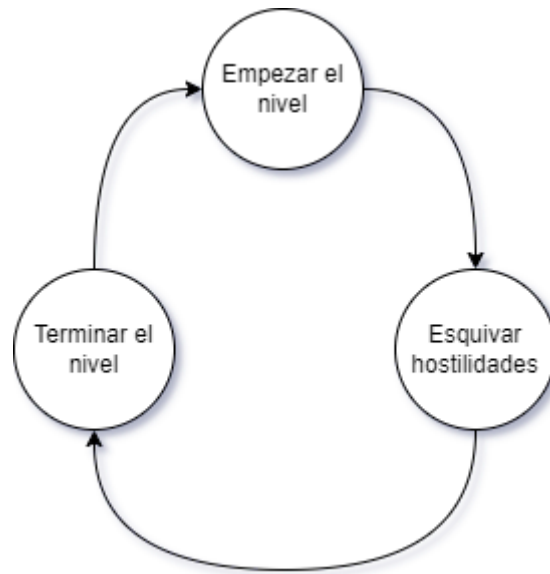
## **1.3 Narrativa**

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 Core Loop**

El *Core Loop* de este videojuego se basará en las siguientes acciones principales: Correr por el mapa (en lo que estaría incluido conseguir vida), esquivar o matar hostilidades (en lo que está incluido esquivar barrancos y saltar plataformas) y llegar al final del nivel. En esta iteración nos centraremos en los tres apartados principales (el loop principal), ignoraremos cosas como enemigos o curas esparcidas por el mapa.



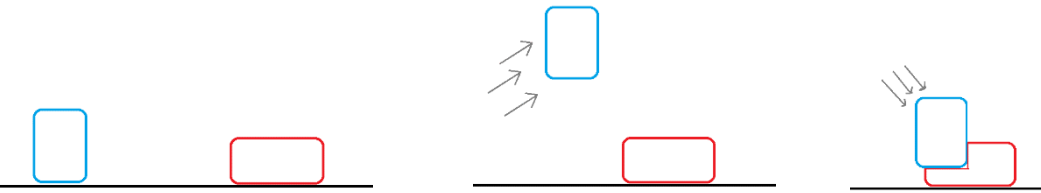
## **2.2 Mecánicas**

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.

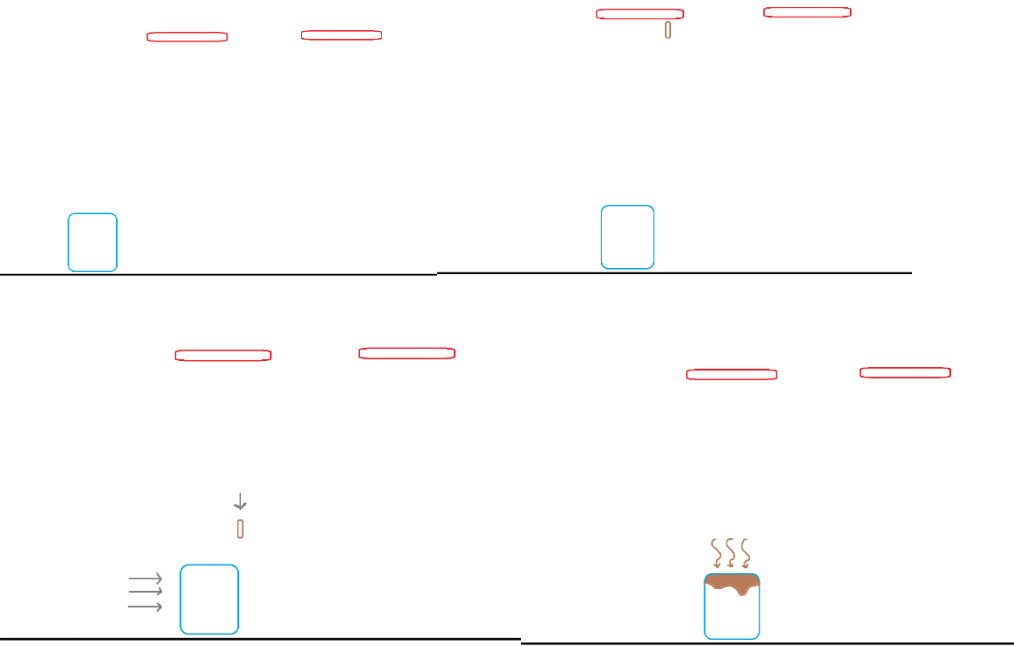


Aquí tenemos otros ejemplos de actividades realizadas en el curso de pasar un nivel:

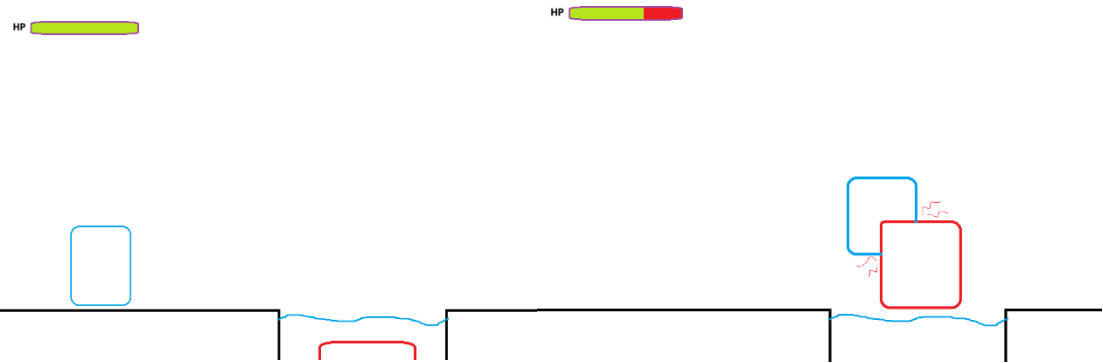
**Salto y eliminación de enemigo en la tierra**



**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 en los 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.

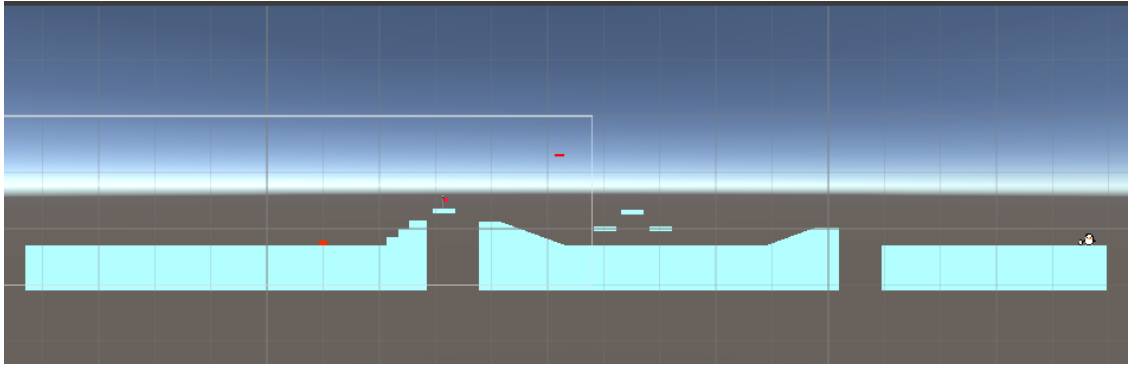
**Orca:** Enemigo que saltará de ciertos barrancos por el nivel. Hará daño en contacto.

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

## 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.

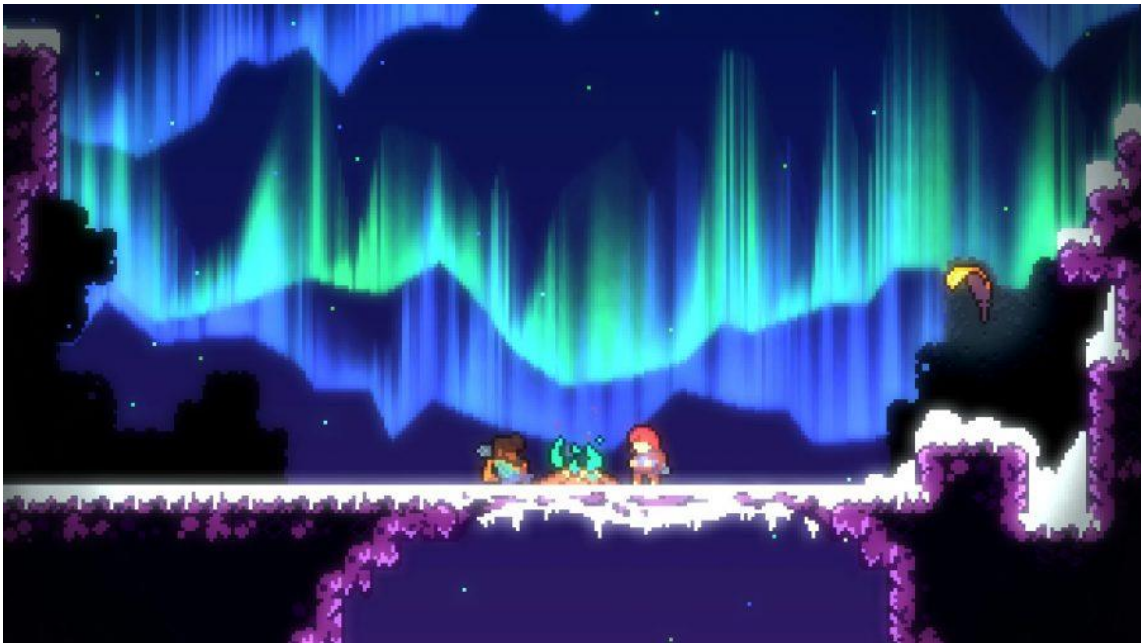


### **3.2 Escenario**

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.

### **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 hacemos evaluaciones y mejoramos aquellos aspectos del juego que no requieran de mucho trabajo (si los hubiera), como audio distorsionado o interfaces borrosas. Estimación de 1 semana.

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.): Álvaro, Rodrigo
- Level designers (encargados de diseñar los niveles de juego y su interfaz): Jorge, Rubén, Gabriel
- Quality assessment: (encargados de evaluar la calidad y jugabilidad del juego) Miguel

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 excell.](#)

[Link al repositorio en GitHub.](#)

# **Retrospectiva**

## **Riesgos**

El mayor riesgo que tenemos ahora mismo es el de no encontrar el balance en cuanto a la complejidad del desarrollo. Lo más complicado que tenemos es el movimiento del jugador. No queremos tener un movimiento muy complejo ya que eso podría ser nuestro mayor retraso. Sin embargo, al ser un juego de plataformas, nuestro mayor foco se debe centrar sobre el movimiento y que quede algo único, pero a la vez responsivo. Hay muchos parámetros a la hora de mover un cuerpo por unity y por ahora nos hemos quedado en algo simple, pero que habrá que ir mejorando.

## **Problemas de retrasos**

El mayor problema que hemos tenido que nos haya retrasado es a la hora de gestionar los conflictos en el git. No entendíamos del todo cómo funcionaban y eso ha provocado que hayamos tenido una gran cantidad de problemas y retrasos.

## **Ayuda al equipo**

Nuestra mayor ayuda y lo que nos impulsa ahora es el saber que cada vez controlamos más y vamos aprendiendo de nuestros errores, como por ejemplo a la hora de manejar el github, ya que estamos usando ramas distintas (cosa que no sabíamos que existía) para no generar conflictos. También estamos encontrando mucha ayuda en internet, como por ejemplo vídeos de youtube o repositorios en GitHub.

## **Meta**

El ideal para esta iteración sería poder lograr ese balance comentado en el primer punto de conseguir que todo se mueva como se tiene que mover. Que quede un juego que sea cómodo de jugar es lo que queremos conseguir para esta iteración.