Game Design Document

RebeKaa

Grupo 5:

Rodrigo Pérez Vilda

Elena De Arriba Jaro

Víctor Martín Aguilar

Sergio Caro Rodríguez

Alejandro Guízar García

Alejandra Del Pino Mortera

ÍNDICE

CONTROL DE VERSIONES

Versión	Entrega	Actualizaciones
Versión 1.0 GDD	21/10/2024	-
RebeKaa 0.0	21/10/2024	-
Versión 2.0 GDD	15/11/2024	➢ Actualización menor en el apartado de mecánica principal, ya que no estaba muy clara la mecánica de la interacción entre los enemigos y la serpiente. ➢ Actualización menor en el apartado de motivación del jugador, ya que no se sabía hasta esta versión la diferencia entre los tres tipos de enemigos. ➢ Actualización menor en el apartado de referencias a otros juegos, ya que el sistema de niveles se ve reflejado en el juego *Super Mario*. ➢ Ampliación de contenido: ○ QA: con los tests realizados de las funciones para comprobar su correcto funcionamiento ○ Documentación software/repositorio: con una breve descripción de los scripts y del contenido del GitHub
RebeKaa 1.0	15/11/2024	➢ Incluida escena definitiva jugable: ○ Ajuste de tamaños de sprites. ○ Mejora de colisiones. ○ Rediseño del fondo para una mejor escala ○ Cambios menores. ➢ Añadidos: ○ Sprite del personaje jugable Kaa. ○ Sprites de frutas. ○ Aparición de frutas de forma aleatoria y cuando muere un enemigo. ➢ Tipos de enemigos: ○ Truncio: lagarto fácil de derrotar. ○ Fénec de dificultad media (sin nombre). ○ Águila de dificultad elevada, enemigo rápido y que solo se puede derrotar con una mecánica específica. ➢ Aparición de enemigos en forma de oleadas, según el tiempo.

		➢ Implementación UI: ○ Barra de vida. ○ Contador de longitud de Kaa. ➢ Mecánica de Kaa: ○ Si choca contra un muro pierde una vida. ○ Si choca contra un enemigo con: ■ Contador mayor: pierde una vida. ■ Contador menor: pierde un segmento de longitud.
Versión 3.0 GDD	13/12/2024	➢ Incorporación del tutorial ➢ Incorporación de los user testings ➢ Incorporación de la retrospectiva ➢ Modificación del apartado “ciclo de juego” añadiendo el diseño molecular ➢ Modificación del apartado “tutorial:mecánica y controles” con las especificaciones de los enemigos y otras modificaciones menores
RebeKaa 2.0	13/12/2024	➢ Implementación de la mecánica para derrotar águilas ➢ Implementación de la pantalla de inicio ➢ Implementación del menú de pausa ➢ Modificación de las oleadas para dificultad y cantidad progresiva ➢ Implementación de las vidas de la serpiente ➢ Implementación de contador de longitud de la serpiente ➢ Implementación de un contador de puntuación que aumenta al comer enemigos ➢ Gestión de errores: ○ Modificación del bug que provocaba longitud negativa al comer las frutas

DESCRIPCIÓN DEL JUEGO

El juego propuesto se basa en un remake del juego “Snake”, en que el usuario tiene el control sobre una serpiente que va eliminando enemigos tratando de no morir en el proceso. Manteniendo del original la movilidad, la vista cenital en 2D, y la mecánica de crecer comiendo frutas. En esta versión aparecerán oleadas de enemigos que habrá que derrotar para obtener una mejor puntuación en función de la velocidad y de los enemigos derrotados. Si pierdes todas las vidas, vuelves a empezar.

TUTORIAL: MECÁNICA Y CONTROLES

La serpiente se mueve en el eje x e y mediante las teclas A, W, S y D según donde se quiera dirigir la serpiente teniendo el plano como referencia, es decir, como el snake original.

En la pantalla aparecerá un contador de la longitud de la serpiente, con el que nos tendremos que guiar para matar a los enemigos, ya que cada enemigo tiene un número asociado que la serpiente tendrá que superar para poder derrotarlo. Además aparecerá la puntuación, basada de momento en la cantidad de enemigos y el tipo de enemigos comidos.

En el caso en el que la serpiente se choque consigo misma o con algún obstáculo, perderá un segmento de su longitud. Sin embargo,si se choca con un enemigo que tiene puntuación mayor que la longitud de la serpiente, el jugador perderá una de las tres vidas con las que cuenta. Al morir en la tercera vida, acaba el juego.

La puntuación de los enemigos es: 3, 5 y 10 de lagartos, fennecs y águilas respectivamente.
Las águilas se podrán derrotar únicamente si se ha tomado previamente la “GreenKaa” lata de refresco que te da alas y hace posible matar a las águilas si se tiene longitud 10 como mínimo.

Para poder aumentar de tamaño, existen las frutas, las cuales irán apareciendo de forma aleatoria en el nivel o aparecen al matar a un enemigo.

Al ser un juego arcade, existirá un contador de tiempo mediante el cual se ganará un bonus de puntuación en base al tiempo empleado por el jugador para pasarse el nivel, de tal forma que, cuanto más rápido, mayor puntuación bonus tendrá.

CONTENIDO

NARRATIVA

Kaa y Rebe son dos serpientes hermanas que vivían felices en la jungla. Pero un día, un monstruo de un vecino desierto llega por la noche, y entre las sombras, secuestra a Rebe. Kaa entonces decide que irá en una misión de búsqueda y rescate de su hermana.

El juego narra como Kaa se debe ir abriendo paso por distintas zonas del desierto, derrotando enemigos hasta llegar a donde se encuentra el Monstruo del Desierto y conseguir derrotarlo y salvar a su hermana.

El monstruo del desierto, debido a las condiciones arduas de su hábitat, ha decidido que va a ser el rey de todo el desierto para que sus esbirros suplan sus necesidades y nunca se muera de sed. Éstos vigilan el desierto y van a buscar sustento para su amo.

El monstruo duerme por el día y sale a realizar sus fechorías por la noche. Es una criatura nocturna por lo que es más poderoso y tiene más energía en la oscuridad. Por ello, la mejor opción de Kaa para llegar hasta el monstruo, será moverse por el día a través del arenoso desierto.
Por ello, cada nivel cuenta con una cuenta atrás. Es el tiempo que tiene Kaa antes de que el monstruo despierte.

Pero hay un problema, Kaa no es autóctona del desierto. Gasta mucha energía en sobrevivir y necesita agua. Por ello, cada vez que derrota a un enemigo, le roba la fruta que éste lleva encima. De esta manera consigue el sustento necesario. Lo que le sobra todos los días, lo va almacenando, como energía de reserva.

En el nivel final, Kaa se enfrenta al monstruo. Sólo si ha conseguido suficiente fruta en los niveles anteriores, tendrá la energía necesaria para derrotar al monstruo y salvar a su hermana. El monstruo, aún de día, es muy poderoso. Si se le acaba el tiempo y llega la noche, el monstruo será demasiado poderoso y no habrá posibilidad alguna.

PERSONAJES

Personajes principales:

- Kaa: es el protagonista del juego, el controlado por el jugador. Es una serpiente que trata de salvar a su hermana de las sucias garras de un monstruo del desierto.
- Rebe (no sale hasta el final): es una serpiente hermana de Kaa, a la que tenemos que salvar en el juego.
- Monstruo del desierto(no sale hasta el final): es el villano principal del juego, raptó a Rebe y Kaa trata de derrotarlo para recuperar a su hermana.

Personajes secundarios:

- Esbirros: son los siervos del monstruo, son a lo que te enfrentas en cada nivel antes del nivel final contra el monstruo (fennecs, águilas y lagartos). Intentarán frenar el paso de Kaa para que no cumpla su objetivo.

JUGADORES OBJETIVO

El juego está pensado para aquellas personas que tengan o sean mayores de 18 años.

MECÁNICA PRINCIPAL DE JUEGO

La mecánica principal del juego consiste en que la serpiente, la cual el jugador controla, se vaya haciendo más grande de forma estratégica a partir de los recursos del nivel, para así poder derrotar a los enemigos de la forma más rápida y eficaz posible.

Para ello, se implementarán varias mecánicas más, como el hecho de que las frutas aparecen en la escena de forma aleatoria o al eliminar un enemigo, con ellas la serpiente aumentará de tamaño. Sin embargo, al ser atacada por un enemigo, la serpiente perderá parte de su cuerpo.

El jugador deberá idear su propia estrategia de juego, saber si le compensa hacerse muy grande a pesar de que eso le limitará el movimiento durante la partida o reservarse las frutas para momentos críticos.

El jugador dispone de tres vidas para lograr superar el nivel, en cada intento la serpiente tiene una longitud, si la longitud llega a 0 pierdes una vida, si pierdes todas las vidas Game Over.

En caso de que la serpiente se choque con una pared o consigo misma, esta perderá una vida.

PILARES DEL DISEÑO

El modo highscore, ya que genera ganas de querer terminar el juego lo más rápido posible para quedar el primero con la máxima puntuación posible.

Las oleadas de enemigos hacen que tengas que distribuir de forma estratégica los recursos aportados al jugador, ya que hay muchas formas de morir sobre todo si la serpiente es muy larga.

La motivación proporcionada por la historia, ya que todo se desarrolla en base a encontrar a la serpiente hermana a la del jugador.

CICLO DE JUEGO

El ciclo de juego se basa en varias etapas:

1. Moverse por el espacio y conseguir frutas

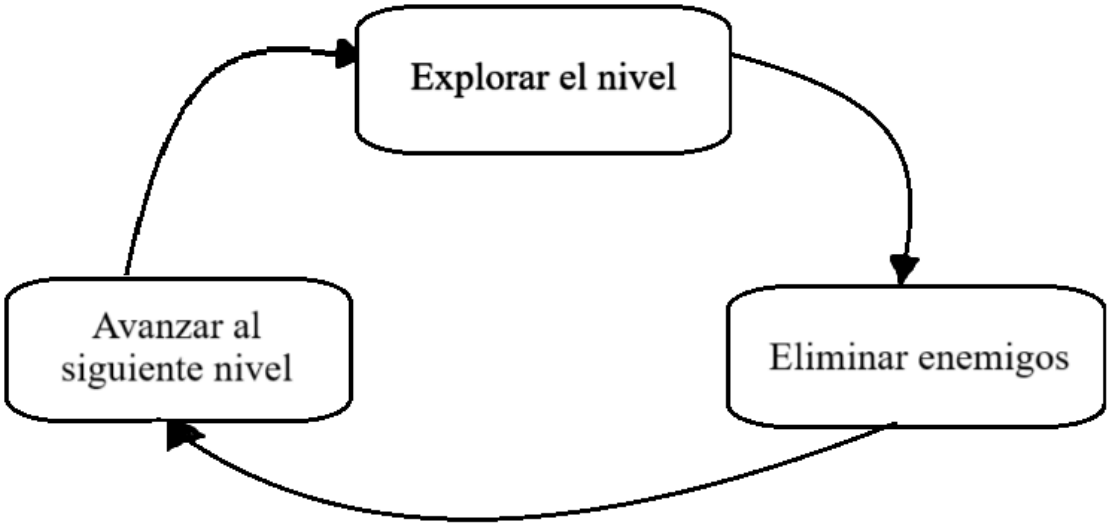
   El jugador podrá moverse libremente por el espacio dado, deberá comer frutas para poder crecer y tener más oportunidades de sobrevivir.

2. Eliminar enemigos

   El jugador deberá de eliminar a los enemigos que se presentan en distintas oleadas para poder superar el nivel

3. Avanzar al siguiente nivel

   Una vez eliminados los enemigos el jugador avanzará al siguiente nivel y vuelta al game loop.



MOTIVACIÓN Y PROGRESIÓN DEL JUGADOR

La principal motivación del jugador será lograr el mejor tiempo posible para obtener la mejor puntuación, añadida a la motivación de derrotar a los enemigos que se presentan, cada uno con su propia característica de movimiento. Como motivación secundaria, el juego cuenta con la historia, que da sentido a los niveles presentes en el videojuego, ya que el jugador encontrará a la hermana de Kaa solamente si se consigue pasar todos los niveles previamente.

FEATURES

- Lograr el mejor tiempo posible para obtener mejor puntuación.

  Cuanto menos tardes en eliminar a todos los enemigos de la ronda, la puntuación obtenida por el apartado de tiempo será mayor. Esto se podrá ver en el marcador de puntuación de la interfaz, además del tiempo tardado. Esta feature se conecta con el pilar de diseño de highscore.

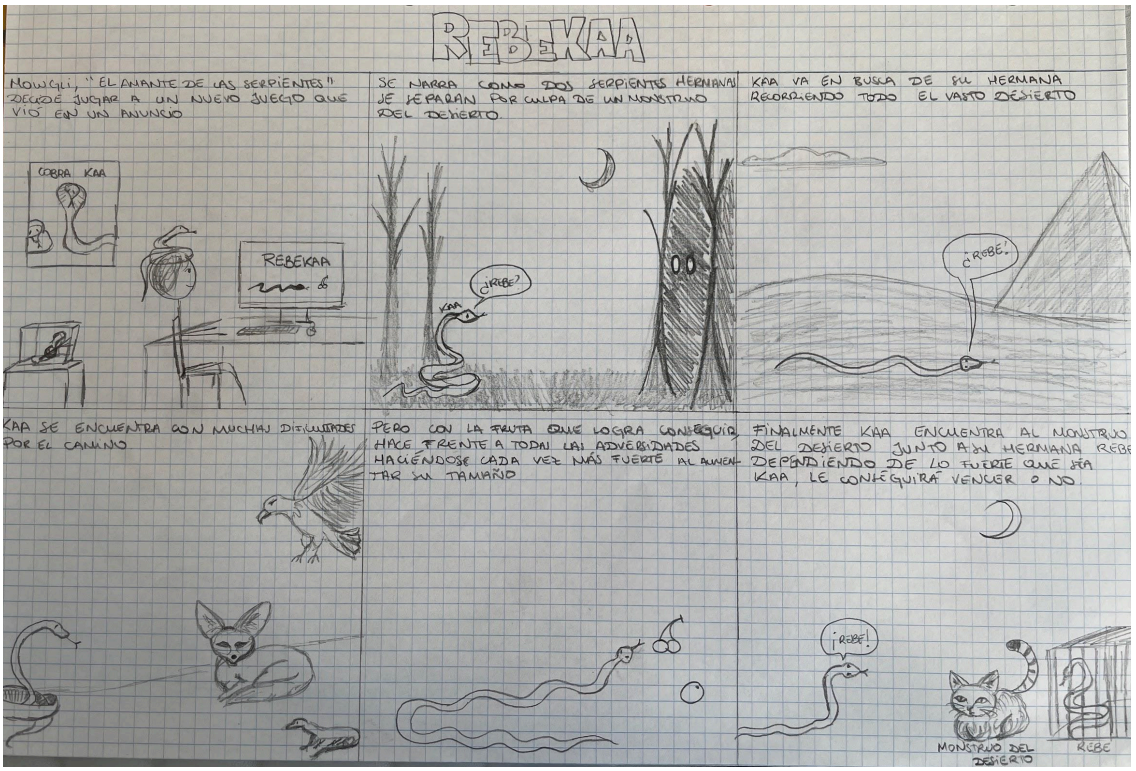
- Sobrevivir a varias oleadas de enemigos de distintos tipos.

  El jugador debe ir eliminando enemigos según vayan apareciendo por oleadas para no verse acorralado y perder la partida. Cuanta más fruta haya comido el jugador, más larga será la serpiente y podrá derrotar a enemigos con un número de poder inferior a esta. Esta feature se conecta al pilar de gestión de recursos y supervivencia.

- Salvar a Rebe, la hermana de Kaa.

  El jugador, que controla a Kaa, tiene como misión salvar a Rebe de las garras del Monstruo del Desierto. Para ello tendrá que hacerse paso por las pantallas de enemigos hasta llegar a su guarida, donde podrá enfrentarse a él. Esta feature está conectada con el pilar de la motivación de la historia del juego.

STORYBOARD



GÉNERO Y REFERENCIAS A JUEGOS PREVIOS

El género principal es arcade. Como referencias, nos hemos inspirado en el juego *Snake*, en los juegos clásicos de *Sonic*, en el minijuego Snake Duel de *Zenless Zone Zero*, y en el minijuego Viaje del Rey de la Pradera incluido en *Stardew Valley*. Además por el modo de puntuación y de aparición de enemigos, hemos tomado como referencia el juego *Vampire Survivors*. También nos hemos basado en el sistema de mundos que tiene lugar en el *Super Mario*, ya que el juego se basa en superar ciertos niveles hasta llegar a rescatar a la hermana del jugador.

QA

User Testing

Todos los datos medidos y analizados se encuentran recogidos en este documento, junto a las conclusiones llegadas en base a los tests:
User Testing

UnitTesting

Para la comprobación del código se ha usado NUnit, una extensión de Unity que permite crear tests individuales. Estos son los tests creados para probar las funciones auxiliares:

RotateEnemyTest()*:* Este test comprueba que se modifique el transform de rotación y de movimiento cuando se llama al método *RotateEnemy()* sobre un enemigo. Consta de 4 pruebas que verifican para cada caso posible si obtiene el valor esperado o no.

ChangeOrientationTest(): Comprueba si sobre un enemigo 2 (águila) se rota el sprite correctamente usando la función *ChangeOrientation()*. Se hacen 2 pruebas, una en caso de no rotar y otra para cualquier tipo de rotación.

SpawnRandomEnemyTest(): Prueba si el enemigo de cualquier tipo se instancie correctamente y se encuentre en uno de los 4 posibles puntos de spawn definidos en la función *SpawnRandomEnemy()*.

SpawnFruitTest(): Verifica si las frutas aparecen correctamente con la función *SpawnFruit()*.

SnakeRotateTest(): Comprueba si la serpiente rota a la izquierda o derecha según su orientación mediante la función *Rotate()*.

SnakeMakeBiggerTest(): Comprueba si se crea un segmento de cuerpo de forma correcta verificando el tamaño de la lista que contiene los elementos de la serpiente tras hacer una llamada a la función *makeBigger()*.

SnakeMakeSmallerTest(): Comprueba si se elimina un segmento de cuerpo de forma correcta verificando el tamaño de la lista que contiene los elementos de la serpiente tras hacer una llamada a la función *makeSmaller()*.

DOCUMENTACIÓN SOFTWARE/REPOSITORIO

- ## Scripts:

  Body: Script correspondiente al manejo de la colisión de los enemigos con el cuerpo.

  Enemy1: Contiene el código referente al primer enemigo, el Lagarto. Esto incluye comportamiento, movimiento y orientación/rotación.

  Enemy2: Contiene el código referente al segundo enemigo, el Águila. Esto incluye comportamiento, movimiento y orientación/rotación.

  Enemy3: Contiene el código referente al tercer enemigo, el Fenec. Esto incluye comportamiento, movimiento y orientación/rotación.

  EnemySpawner: Código referente a la invocación de enemigos en pantalla por oleadas en cada ronda. Como entrada tiene los prefabs de los enemigos a invocar.

  FruitSpawner: Código referente a la invocación de frutas en escena en intervalos de tiempo aleatorios.

  Snake: Contiene el código de la serpiente, que incluye el movimiento, la mecánica de aumentar su longitud al comer frutas, la mecánica de derrotar enemigos, y la mecánica de perder vidas.

  GreenKaa: Contiene el código referido a la bebida nombrada como “GreenKaa” en el que se trata el spawneo de la misma

  PauseManager: Contiene el código referido al menú de pausa y su funcionamiento según lo que el usuario elija, si reanudar, salir…

  Buttons: Contiene el código referido a los botones de la pantalla de inicio.

- **Repositorio:**

(Los contenidos entregables y definitivos se encuentran en la rama main del repositorio, el resto de ramas son ramas de progreso individual o contenedores de sprites que se usan para construir progresivamente la rama main.)

Ejecutable: Contiene el ejecutable del juego para su prueba.

Excel de flujo acumulado: Archivo .txt con enlace a dicho excel.

GDD: PDF actualizado del Game Document Design.

PlanificaciónNYTrello: PDF de la Planificación de la N-ésima iteración con enlace al Trello incluido al final del documento.

RebeKaa: Carpeta que contiene el Proyecto Unity actualizado.

RETROSPECTIVA

¿Qué nos retrasó o nos hizo perder el control?

En primer lugar, al igual que en la iteración anterior, los exámenes y otras responsabilidades académicas como proyectos de otras asignaturas. Es muy complicado ponerse de acuerdo si los miembros no coinciden con las asignaturas, ya que mientras algunos están en algún momento concreto libres, los demás están ocupados, o incluso hay momentos en los que ningún miembro tiene hueco para trabajar en el videojuego. Pese a esto, como pasó lo mismo la iteración anterior, esta vez se ha mejorado la comunicación grupal y la organización, de tal forma que en todo momento se sabía quién hacía qué cosas. Otra cosa que nos retrasó fue la confusión de mecánicas relativas al juego, ya que había confusión de la longitud necesaria para derrotar a cada enemigo, de la forma de derrotar a las águilas y bastantes cuestiones más de este tipo, lo cual causó conflictos entre los miembros ya que no nos poníamos de acuerdo con algunas medidas como por ejemplo, la cantidad de enemigos por oleadas u otras cuestiones.

¿Qué nos ha impulsado en este sprint?

Este sprint nos ha impulsado la idea de tener un juego sin bugs y más jugable, con su pantalla de inicio, su menú de ajustes y su implementación completa de enemigos, intentando en la medida de los posible evitar los posibles bugs que se puedan generar en su programación. Aparte de eso, el hecho de que haya usuarios que lo vayan a probar, ya que se ha querido mostrar un buen videojuego dentro de lo que cabe y del tiempo dedicado al mismo. Todo esto ha sido lo que en gran parte nos ha impulsado en este sprint.

¿Qué nos hará ir más rápido en el siguiente sprint?

Dado que hemos encontrado muchos bugs a pocos días de la entrega, nos centraremos en ellos para tener un juego jugable y divertido, por lo que descartaremos la idea del boss final para poder ir más rápido y arreglar todo lo que tenemos que arreglar de lo ya hecho. Además, durante estas iteraciones surgieron dudas relativas a las mecánicas de enemigos que no terminábamos de tener claras, pero durante esta iteración se han resuelto todas esas dudas y además se ha implementado por lo que no creemos que eso nos retrase mucho este sprint.

¿Hubo algún factor que afectó la productividad de este sprint?

Como se ha mencionado antes, la cantidad de trabajo que exigen las otras asignaturas fue un factor que nos retrasó bastante, aunque si que hubo personas que trabajaron durante ese tiempo, otras se dedicaron a las otras asignaturas, retrasando de ese modo este proyecto. Otro factor que afectó a la productividad fue el hecho de dejarlo todo para el último día, hubo integrantes del equipo que no trabajaron casi hasta los últimos días, retrasando por el ejemplo el trabajo de QA, pese a todo se ha intentado salir adelante, aunque apretando mucho. Afectó también el desconocimiento de algunos de los miembros acerca de git, generando algún que otro problema a la hora de subir o hacer merge de las ramas.

¿Qué nos preocupa?

Nos preocupa la cantidad de bugs que se han presentado durante esta iteración, debido a los cuales nos vemos obligados a retirar la idea del boss final dado que no tenemos tiempo suficiente visto lo visto. Dada esta decisión, nos preocupa no poder dar a los usuarios esa jugabilidad y diversión que queremos que tengan mientras juegan al videojuego, el cual planteamos para pasar el rato, como cualquier otro del género arcade.

ORGANIZACIÓN DEL EQUIPO

- **Team Lead:**

  Sergio Caro: Diseño del menú de pausa y realización de un user test.

- **Programmers:**

  Víctor Martín: Implementación del menú de pausa, implementación de la mecánica de derrotar águilas e implementación del contador de longitud.

  Elena De Arriba: Realización de un user test, diseño de sprite de vidas infinitas y arreglo de bugs de cerezas que dan puntuación negativa.

- **Level designers:**

  Alejandro Guízar: Diseño de la puntuación, diseño pantalla de inicio, diseño del resto de oleadas, diseño del sprite de alas, revisión de tareas y creación del ejecutable.

  Rodrigo Pérez: Implementación pantalla de inicio, realización de dos user tests, diseño e integración del menú de pausa, unificación de programación en rama main y diseño de los botones del menú de pausa.

- **Quality Assessment:**

  Alejandra Del Pino: Coordinación y unificación de los user testings, revisión y actualización del GDD, planificación 4, retrospectiva, seguimiento del diagrama de flujo acumulado, seguimiento del Trello, revisión de tareas, documento user testing, realización de siete user test, diseño del resto de oleadas y diseño molecular.