

# RECINTO SANTIAGO

**Asignatura:**

Programación de Videojuegos

**Tema:**

Proyecto*Dark*

**Presentado por:**

Carlos Fabián Arias Guzmán 1-18-9177

Bryan Marc 1-19-0040

Yves Gervens Constant 1-18-2068

**Profesor:**

Iván Mendoza

**Link:**

[**https://github.com/gh4m21/dark-proyectofinal**](https://github.com/gh4m21/dark-proyectofinal)

**Fecha:**

“” de diciembre de 2022

**INTRODUCCIÓN**

En este proyecto se establecerán los diferentes tipos de elementos que tendrá a lo largo de su desarrollo, además de sus objetivos tanto generales como específicos, sin olvidar de dónde proviene la idea, su conceptualización, detalles del arte y las diferentes herramientas que hizo posible su realización.

Es importante tener en cuenta el tiempo y la disponibilidad de cada uno de los integrantes que se comprometieron al máximo para el desarrollo de este proyecto, además de la compresión e investigación de cada una de las mecánicas implementadas. No cabe duda que fue un camino de entendimiento y error, en el cual dio su fruto, la adquisición de conocimientos.

Este proyecto marca un principio, pero no un final para aquellos que han sido más que motivados a seguir aprendiendo de las diferentes mecánicas y funciones de esta rama de la programación. Es más que un arte, en el cual puedes interactuar y sumergirte en un mundo de posibilidades infinitas, teniendo como limitante, tu imaginación.

**INDICE**

# CAPÍTULO I: VIDEOJUEGO Y HERRAMIENTAS DE DESARROLLO

1.1 Descripción

1.2 Motivación

1.2.1 Originalidad de la idea

1.2.2 Estado del Arte

1.3 Objetivo general 1.4 Objetivos específicos

1.5 Escenario

1.6 Contenidos 1.7 Metodología

1.8 Arquitectura de la aplicación

1.9 Herramientas de desarrollo

# CAPÍTULO II: DISEÑO E IMPLEMENTACIÓN

2.1 Planificación (Diagrama de Gantt)

2.2 Diagramas y Casos de Uso

2.3 Plataforma

2.4 Género

2.5 Clasificación

2.6 Tipo de Animación

2.7 Equipo de Trabajo

2.8 Historia

2.9 Guion

2.10 Storyboard

2.11 Personajes

2.12 Niveles

2.13 Mecánica del Juego

# CAPÍTULO III: DESARROLLO

3.1 Capturas de la Aplicación (Documentación completa del desarrollo, Scripts, Sprites, Prefabs e

imágenes)

3.2 Prototipos

3.3 Perfiles de Usuarios

3.4 Usabilidad

3.5 Test

3.6 Versiones de la Aplicación

# CAPÍTULO IV: PUBLICACIÓN

4.1 Requisitos de la instalación

4.2 Instrucciones de Uso

4.3 Bugs

4.4 Proyección a Futuro

4.5 Presupuesto

4.6 Análisis de Mercado

4.7 Viabilidad

**CONCLUSIONES**

**REFERENCIAS BIBLIOGRÁFICAS**

**CAPÍTULO I: VIDEOJUEGO Y HERRAMIENTAS DE DESARROLLO**

# 1.1. Descripción

El proyecto que se va a ilustrar y desarrollar es un concepto diferente a lo que conocemos de la jugabilidad en 2D. Resulta que siempre asumimos que si un juego esta caracterizado, ya sea un arte pixelado o en dos dimensiones automáticamente lo relacionamos con un juego de no tan alta calidad. En este proyecto, el videojuego va utilizar 2.5D, quiere decir que se van a utilizar tanto el 2D y 3D. Creando así una jugabilidad de 3 dimensiones con objetos, animaciones, NPCs, enemigos, y múltiples elementos del escenario que estarán completamente en 2D.

Esto da una jugabilidad bastante atractiva, ya que se da a relucir la combinación de ambas dimensiones y la utilización de distintos elementos, como las luces, partículas, difuminación que combinados dan un resultado de inmersión total.

Se han utilizado diferentes herramientas para dar vida a nuestro personaje y los diferentes elementos que se encontrarán en el mapa, que estará más destallado a lo largo de este proyecto en la sección de *Herramientas de desarrollo,* en donde se darán a conocer cada una de estas para el desarrollo de nuestro videojuego.

Ahora bien, con la combinación de código y las diferentes herramientas a utilizar, hemos desarrollado un juego llamado “*Dark*”, el cual utilizará 2.5D para su jugabilidad e ilustración.

Una descripción breve de nuestro videojuego, es que trata de un caballero de la orden de la luz, el cual desafiará a las almas de las sombras o “sombras”, para recuperar la luz de la ciudadela. Cabe destacar que la dificultad aumentará y los enemigos serán aún mas fuertes cada vez que se pase un nivel.

Los controles serán muy simples y además el juego contará con un breve tutorial para la explicación de sus mecánicas al inicio del primer nivel a modo de tutorial.

# 1.2. Motivación

Lo que nos motivó a la creación del videojuego fue la utilización de formato 2D y el 3D además de todas las funciones que estos ofrecen, como las partículas, la utilización de luces y sombras, y las diferentes mecánicas que este juego goza las cuales explicaremos a detalle más adelante.

Además, como parte de motivación, nos ayudó a explorar más los conceptos e indagar nuevas técnicas para el desempeño del personaje, como la utilización de nuevas herramientas en el proceso del desarrollo. Fue toda una experiencia la cual saboreamos con nuevos conocimientos adquiridos.

Con este proyecto queríamos devolverle la chispa de las entregas de los 90, con sus característicos y nostálgicos 32 bits, además de sus bandas reconocibles hasta hoy en día, queremos darle un ambiente un poco más fresco e ilustrar luces y sombras al videojuego.

## 1.2.1. Originalidad de la idea

Este videojuego está ambientado y motivado en el videojuego Octopath Traveler. El cual utiliza una gran cantidad de elementos para combinar las luces y los objetos en 2D en un mundo 3D.

Octopath Traveler (imagen de la izquierda) es un juego RPG japonés, el cual busca dar conocer las diferentes fortalezas y nostálgicas inmersiones de los videojuegos de 32 bits.



Hemos utilizado el concepto de Live a live (imagen de la derecha) un juego con el concepto de Octopath Traveler, pero este relaciona diferentes épocas las cuales pueden ser la japonesa con los ninjas, la actualidad, far west, entre otros. Ya que en Live a Live no se encuentra la época medieval decidimos ambientarlo a esta época. El juego constará de ciertos niveles para derrotar a las sombras para ir aumentando la luz en el mundo medieval.

Originalmente Octopath Traveler es un juego RPG, pero con jugabilidad por turnos en cuanto al combate se refiere, en este proyecto no utilizaremos este sistema de combate, sino que optaremos por un sistema de combate dinámico, en el cual los enemigos atacarán simultáneamente y nosotros como jugadores debemos de desarrollar estrategias para poder defendernos de la ofensiva enemiga y poder pasar el nivel.

En los niveles existirán objetos rompibles los cuales nos incrementarán la vida o nos darán monedas de oro, las cuales podrán ser canjeables por más vida, pociones, objetos para aumentar el ataque y la defensa, entre otros.

Este juego es semi mundo abierto, ya que en la ciudadela principal podemos ir a las misiones, visitar la tienda para aumentar la vida o la herrería para obtener objetos los cuales nos aumenten algunas características del personaje.

## 1.2.2. Estado del Arte

En general estamos creando diferentes bocetos de cómo serán las diferentes mecánicas del videojuego, además el diseño utilizando Pixel Art para la ilustración del personaje como los demás elementos. En el apartado de *Herramientas de desarrollo* se mencionarán alguna que otra para la elaboración de estos elementos.

# 1.3. Objetivo general

Con este proyecto se quiere dar a ilustrar y desarrollar un videojuego pixelado utilizando mecánicas de luces y sombras para su ambientación.

No es un secreto que, con la realización de este proyecto, obtendremos más conocimientos a la hora de la realización e investigación de cada una de las pautas para su correcta entrega. Gracias a estos conocimientos podemos desenvolvernos aún más rápido y ser más óptimos en la identificación de erros o bugs en nuestro proyecto.

**1.4 Objetivos Específicos**

Entre los objetivos específicos que podemos encontrar, están los siguientes:

* El tipo de Lucha Live to Live
* Desarrollo e ilustración de un videojuego pixelado
* Volver al estilo de videojuegos de los 90s
* Tener un sistema de combate dinámico

**1.5 Escenario**

En cuanto al escenario será elaborado sobre un mundo semi abierto donde haremos uso del arte y escenas de la época medieval, solo haremos uso de esta para marcar la diferencia.

**1.6 Contenidos**

Nuestro videojuego debe tener actualizaciones notables de manera constantes y buena jugabilidad para los usuarios, esto hará que el coste aumente.

# 1.7. Metodología

La metodología que vamos a utilizarse para acabar con este proyecto es la metodología en cascada que consiste con un marcado de inicio y fin. Es un proyecto que vamos a desarrollar desde cero y con grandes posibilidades de ampliación, es la razón porque utilizaremos esta metodología agiles para poder modificar en los sprints los distintos niveles, la historia, los movimientos, el guion entre otros. Además, permitirá a los demás integrantes de este proyecto de trabajar de forma simultáneos y con periodos

Una particularidad de esta metodología es que podemos usarla de forma cascada en combinación con los “sprints” para las diferentes entregas.

**Cascada:**

El modelo en cascada es un modelo en la que el proceso de desarrollo se hace de forma lineal las distintas partes del proyecto hasta llegar al fin. Esos distintos partes son: Análisis, Diseño, Implementación, Verificación y Mantenimiento.

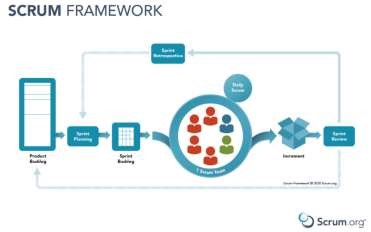


**Figura: Cascada**

*Fuente: https://crehana-blog.imgix.net/media/filer\_public/00/ca/00ca80f8-9af9-4886-8f41e8bdeb8f26d6/modelo-en-cascada-como-funciona.jpg?auto=format&q=50*

**Scrum:**

Vamos a utilizar la metodología SCRUM para realizar una serie de iteraciones a través de las cuales se va a completar el proyecto etapas a etapas para llegar hasta el producto final.



**Figura: Scrum**

*Fuente: https://proactiveoffice.com/wp-content/uploads/2021/02/Scrumframework-1024x639.png*

# 1.8 Arquitectura de la aplicación

El juego Dark se desarrollará utilizando Unity que permite ser multiplataformas y instalado a cualquier tipo de dispositivo.

La arquitectura de la aplicación se plantea de la siguiente forma:

El juego habrá varias escenas que formaran parte de la interfaz (UI) es decir el menú, las configuraciones, la pantalla de menú de niveles, la pantalla de instrucciones. Cada nivel va a tener una escena donde se diseñar los prefabs y Sprite de los elementos distintos del juego, es decir el personaje, los terrenos, las armas, etc.

El juego se contará con un personaje principal y varias escenas donde el personaje principal hará parte principal de esas escenas. Este personaje seguirá por una cámara y con botones para dirigirla y atacar los enemigos.

El juego tendrá un game manager para procesar las escenas y el menú y tener las configuraciones del sistema de juego es decir los audios, etc.

Un elemento importante de este tipo de arquitectura son los scripts que es la lógica del juego es decir que permitirá procesar el movimiento del personaje principal, los ataques del personaje y enemigos, calcular los puntos de vida entre otros.

# 1.9 Herramientas de desarrollo

Las herramientas de desarrollo para realizar este videojuego son de todos tipos y vamos a desarrollarla a continuación:

* Unity 3D/2D: El videojuego se desarrolla en un ambiente 3D con personajes y objetos de 2D y es un videojuego de tipo 2.5D. Se elige este motor gracias a los conocimientos adquirido en la asignatura “programación de videojuegos”. Este motor tiene muchos elementos importantes y librerías para facilitar el desarrollo de un videojuego.
* Lenguaje de Programación C#: Elegimos este lenguaje porque permite usarla con Unity para tener acceso a los elementos nativos de los soportes como el teclado. Este lenguaje es ideal y es adaptado para los videojuegos porque aumenta las actuaciones del videojuego.
* Entorno de desarrollo Visual Studio y Visual Studio Code: Estos son unos de los IDE más completadas del mundo y es ideal para escribir los lógicos del videojuego.
* Gráficos: Photoshop, aseprite, pixelart, Gimp, Tilemap: Utilizaremos esas herramientas de acceso libres para poder crear los elementos gráficos como sprites, terreno y los demás elementos gráficos.
* Lightrays-2d-effect: Utilizaremos estos paquetes libres del assets store para poder crear efectos en el entorno del juego.
* Audio Freesound, indie game music, free sound projects: Utilizaremos los elementos y efectos sonidos audio de esos sitios webs para poder generar sonido para el videojuego Dark.
* Documentación: utilizaremos los paquetes de Office para escribir la documentación del juego como planificación.
* Gestión de proyecto, Diagrama de Gantt: utilizaremos el diagrama de Gantt para poder dividir el proyecto y asignar las tareas.



**Figura: Herramientas**

**CAPÍTULO II: DISEÑO E IMPLEMENTACIÓN**

**2.1 Planificación (Diagrama de Gantt)**

**2.2 Diagramas y Casos de Usos** A picture containing graphical user interface

Description automatically generated

Diagram

Description automatically generated

A screenshot of a video game

Description automatically generated with medium confidenceDiagram

Description automatically generated with medium confidenceDiagram

Description automatically generatedGraphical user interface, application

Description automatically generated

**2.3 Plataforma**

Nuestro proyecto será basado para ser consumido en PC, por el motivo de que la jugabilidad será mas fluida y con mayores gráficos, pero puede expandirse a otras plataformas en un futuro, Dark no tiene límites.

**2.4 Genero**

Dark será desarrollado como un juego de Rol (RPG), es un género de videojuegos donde el jugador controla las acciones de un personaje inmerso en algún detallado mundo.

**2.5 Clasificación**

Dark es un juego que implica consciencia de parte del jugador por ello decidimos clasificarlo en B15,

Clasificación “B15”: Contenido para mayores de 15 años.

**3.3 Perfiles de Usuarios**

- Personas mayores de 15 años que tengan un alto consumo de videojuegos

- No requiere una alta formación educativa

- Capacidad de razonamiento lógico

**3.4 Usabilidad**

- El usuario puede aprender a utilizar la interfaz del juego de una manera intuitiva y fácil.

- Proveer facilidad para que los jugadores alcancen y entiendan su objetivo en el videojuego.

- Ser fácil de recordar su funcionamiento para futuras visitas.

- Colores visualmente agradables para el usuario