My Damnation

Hernandez Monroy Luis Fernando

Documento de diseño

My Damnation

Este es un proyecto que momentáneamente no cuenta con un nombre estable más que uno clave para referirse a él, este juego está clasificado de acuerdo al sistema ESRB (Entertainment Software Rating Board), “T” para adolescentes y mayores de edad, ya que contiene material sugestivo, comedia vulgar, violencia y uso sugestivo de alcohol.
Este juego es una sátira de un dungeon crawler, rogue-lite, con temática religiosa, satírica a varios temas que son abarcados en múltiples documentos, como principal inspiración y referencia de La divina comedia de Dante Alihieri, esto no es una adaptación, solo como referencia para el juego es utilizado este material y combinado con otros medios de información.

One Shot

Eres un caballero de la edad media, trabajando para la realeza, está todo subdividido por clases sociales y hay desde clases marginadas hasta los burgueses que son codiciosos, siendo un caballero de la realeza, no puedes opinar respecto a cómo se encuentra el reino, el rey desata guerras para adquisición de bienes, su descendiente es el mejor espadachín de la realeza, un verdugo que aprovecha su posicionamiento dentro del castillo, un mago deseoso de conocimiento sin temor a los bordes de lo bueno y lo malo, el caballero, todos son enviados a una guerra, sin embargo es sobrestimado el enemigo y todos mueren, tú siendo el caballero, llegas al infierno, donde encuentras un alma en el purgatorio, te habla del régimen del infierno y como está subdividido por los siete reyes del infierno, sin embargo también te hace saber que es imposible escapar, ya que el infierno siempre está en constante cambio y una vez que eres atrapado por las flamas (muerto) nunca puedes regresar a tu lugar de origen, tambien hace mencion de que es un espacio atemporal por lo que veras cosas extrañas dentro que no tendrá, la más mínima coherencia, sin embargo debes de aprovechar todo lo que encuentres, ya que a modo desobediente te aventuras a averiguar mas del infierno e intentar encontrar a los siete reyes, para conseguir respuestas de cómo puedes regresar.
Cada rey es representante a un pecado capital y cada nivel tiene su diseño, por lo que es posible darle su personalidad a cada una de las secciones del juego, y como previo mencionado, ya que es un espacio atemporal, se puede ingresar multiples armas como referencia a distintos juegos o a la vida real, a la era moderna, sin esperar mucho sentido de esto. También se espera poder agregar un octavo piso que sea representativo del “averno” y posteriormente dos más que seran “Paraiso”

My Damnation

Especificaciones Técnicas e interactivas

Single player.
Juego 2D, tercera persona, perspectivas isométricas, caballero y 3/4.
Diseño: Pixel art.
Lenguaje de programación: C#
Hardware: Inicialmente ordenadores, portátiles y de escritorio.
Compatibilidad: Soporte de Control(Óptima jugabilidad) y teclado.
Software: Unity 2018, Krita, Garageband.
ESRB Teen
Venta completa del juego, dependiendo su éxito puede haber actualizaciones gratuitas para avivar la comunidad.

Planteamiento de problemas;

1.- Sistema de vida.
2.- Sistema general de navegación.
3.- Mitigación daño (esquive) y Ataque.
4.- Recibir daño.
5.- Reconocimiento de ítems.
6-. Sistema de inventario.
7.- Menú principal.
8.- Menú con distintos slots de carga de partida.
9.- Interacción con NPC.
10.- Sistema de diálogos con NPC.
11.- Sistema de combate.
12.- Daño a los enemigos.
13.- Daño recibido.
14.- Eliminar objetos destructibles y enemigos.
15.- Detección de teclado o control.
16.- Inventario desplegable.
17.- Pantalla de muerte.
18.- Decisiones en el juego.
19.- Boss multifacetas.
20.- Pausa.
21.- Procedural dungeon

My Damnation

Procedural Dungeon

Primero expondré de este contenido para que se entienda más el concepto del juego, sin embargo es el contenido más complicado de los previos abarcados.
Hay dos tipos de soluciones para esta abstracción de concepto:
Dungeon Crawler o Drunken walk, ambos son soluciones abstractas y con sus beneficios y problemas.

Dungeon Crawler

Este método es similar a crear una mazmorra similar a “the legend of zelda” donde las medidas ya están definidas por cuartos, sin embargo el factor procedural entra en que tu creas múltiples habitaciones y la orientación o el acomodo de ellas siempre será diferente a partir de una selección en un índice, es decir, uno tiene que diseñar los props y posterior las combinaciones dentro de habitaciones e indexados, para dejarlo al programa jugar con los números aleatorios. Incluso el acomodo de los enemigos, uno debe pre posicionar una posible estancia al entrar al cuarto, para evitar que se posicionan sobre los props o donde no deberían.

Drunken Walking

Este método es más impredecible, sin embargo se obtienen resultados bastante interesantes, ya que se programaría un tile que se mueva en una dirección aleatoria por un número de píxeles, un cierto número de veces, también se le debe asignar que al ir en una dirección que ya ha pasado, no lo cuente cómo pasó, así obtienes un resultado bastante impredecible pero una mazmorra de un piso con pasillos y zonas más naturales, las cuales se programa para asignar automáticamente los tileset correspondientes a las orillas o esquinas.

Sin embargo todavía no tengo una resolución clara de que es lo que podría hacer con cada uno o cuál es el resultado final del proyecto, que deseo alcanzar, por ello expongo primero esta problemática sin solución.

My Damnation

Sistema de vida

El usuario necesita una barra de vida que represente el valor igual a su vida, su mayor vida posible es 100 como límite, sin embargo con las upgrades que creas en el futuro, este valor se puede multiplicar para que aumente o reduzca, sin desplegar una diferencia visual en la barra de vida.
Nota : Se empieza pruebas con valores enteros ya que el damage puede ser cuantificado en enteros o flotante, pero es cuestión de prueba y ver que acomoda más a valores ya integrados.

Pseudocódigo

Declaración de variables

Int maxHealth con valor de 100
Int curHealth sin valor
txt healthCount (Esto recibirá el valor de la curHealth, transformándolo en un string posteriormente).

Int daño con valor 0

Algoritmo

Se inicializa curHealth con valor a 100

La barra de vida estará desplegada siempre en el costado izquierdo superior, de manera horizontal, con el valor desplegado curHealth, que se actualizará constantemente, y variara el UI de la barra dependiendo el monto de vida que se tiene, su máximo valor posible desplegado es de ”100”, por lo que cualquier otra entrada mayor a ella, no es desplegada, si no hasta posteriormente que se consigue alguna mejora de equipamiento y aumenta la vida.

Si curHealth es mayor a maxHealth, entonces curHealth sera igual a maxHealth

Si el box collider hace contacto con el target “enemigo” o “ataqueE”.
Obtener los componentes del “enemigo” o “ataqueE” , encontrar el valor de daño
Operación algorítmica curHealth= curHealth - daño
Imprimir curHealth

Fin del Algoritmo

Diagrama de flujo

Movement System

El sistema de movimiento será en 360 grados el los dos Axis que jugará (X, Y), para dar mayor fluidez al movimiento del personaje, si se usa el keyboard, se moverá en total en ocho direcciones cada una determinada por los puntos cardinales.

Gracias al rigidbody podemos construir físicas donde el jugador empiece desde un estado inicial y cuando llegue a su estado final, desacelerar que simula el "momento" adquirido con la velocidad.

Pseudocódigo

Declaración de variables

Se declara un Enum público que posteriormente se usará en el código para el ataque, esquive y otras variables
Declaración del estado público del player caminando
Variable flotante maxSpeed con valor a cero
Vector3 transform
Rigidbody

Inicio del algoritmo

Iniciar el animador obteniendo las propiedades de la animación correspondientes.

Declaramos la traslación inicial es cero.
Posterior obtener los componentes transform y cambiar de posición del player por medio de input getAxis, con las teclas asignadas al movimiento horizontal y vertical.

Transformar posición sumando traslación por máxima velocidad por delta time.

Por medio de If, si la traslación no es igual a su valor original, cero(Velocidad inicial)

set booleana para para animar caminata true
Activa la animación correspondiente a dirección de traslación

else la velocidad vuelve a cero, setearla false (asi cambia de animación siempre estando en el estado “caminando”)
Fin del algoritmo

My Damnation

Mitigación de daño (esquive) y Ataque

Esto servirá para que el jugador tenga una manera de no recibir daño al moverse, con una penalización post realizado el esquive, de unos unos segundos.
Este código pertenece a la misma categoría de movimiento, por lo que solo se agregaran los cambios que se hacen dentro de la declaración y dentro del código.
Ambos serán comparados con el state “caminando” para evitar que se pueda cambiar de estado por frame entre ataque y esquive usando los a la vez.

Pseudocódigo

Declaración de variables

Agregar un enum al listado para la mitigación y ataque.

Algoritmo

Utilizar un transform para modificar la velocidad normal del movimiento y multiplicar la por 1.5, para esquivar más rápido.

Usar una variable If para decir, “si se está caminando y es presionado el botón correspondiente a esquivar” entonces activar la animación de esquivar.

Definir que cuando se esquiva, se obtienen los componentes del sistema de vida, después si el target tiene el tag de “enemigo” , “ataqueE”, y el estado actual del player es esquive, que el daño sea cero.
Regresar el estado a caminar y definir un tiempo para evitar que se pueda esquivar siempre.

Fin del algoritmo

My Damnation

Recibir daño del enemigo y hacer daño

El daño será variable dependiendo al arma que se use en ese instante, la mejor manera de utilizarlo ahora mismo en con un valor entero equivalente a 10, para objeto de prueba y cinco para otro ataque de prueba.

Pseudocódigo

Algoritmo (Ataque previamente establecido en el diagrama de flujo anterior)

Se activa la animación de ataque y la boxCollider de la animación, correspondiente a la dirección que te encuentres observando.
Al acabar la animación y regresar al estado caminando
Enfriamiento de la animación por .3 segundos(valor de prueba)

Fin del algoritmo 1

Algoritmo 2 (Enemigo vida)

Declaración de vida sin valor
valor entero vidaActual = 0
valor entero dañoContacto;
valor entero dañoAtaque;
Box Collider dentro del enemigo prueba

Inicializar la vida con valor igual a 20.
Obtener los valores del boxCollider.
vidaActual = Vida

Si BoxCollider se encuentra con target “ataque”
Obtener los valores del objeto con target ataque(player)
vidaActual = vida - daño (del objeto con target ataque)
Impresion de vidaActual
Fin del algoritmo 2

My Damnation

Seguimiento de cámara.

La cámara seguirá al jugador y tendrá un límite para no poder observar fuera de los bordes, así no tendremos que usar tantas texturas para tapar los bordes.

Pseudocódigo.

Algoritmo
Declaración de las variables con valor flotante, para un máximo y un mínimo, también una variable que seguirá al personaje.

Creamos un target que siga al personaje a donde vaya en el eje Z

Calculamos la esquina en el eje X y posteriormente en el eje Y (Máximo y mínimo)

Aplicamos el movimiento de la cámara suavizado.

Fin del algoritmo.

My Damnation

Destrucción de objetos (enemigos y objetos).

Este código ayudará a destruir los objetos que porten el mismo, por medio de comparación de tags.

Pseudocódigo

Algoritmo

Creamos una función que pueda ser llamada por cualquier consigo

Esta función contendrá destruir el objeto

También lo desactiva de la escena para mejorar el procesamiento de la memoria.

Fin del algoritmo.