



Universidad del Valle

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas y Computación

Computación Gráfica

Proyecto Final:

The Legend of Zelda

Presentado por:

Sergio Ortiz Paz

Código:

1526478

Docente:

Carlos Andrés Delgado Saavedra

Febrero – Junio 2015

INFORME DE PROYECTO FINAL COMPUTACIÓN GRÁFICA

Introducción

El famoso juego The Legend of Zelda: A Link to the Past para la consola Super Nintendo es un video juego de acción y aventuras (Rol) creado por la compañía Japonesa Nintendo. En el cual el protagonista/héroe conocido como Link debe de enfrentar en batalla a varios monstruos entre ellos Ganon para rescatar y proteger a la princesa Zelda y demás doncellas del reino.

El proyecto trata sobre la implementación de una pequeña parte de este video juego mediante la realización de 2 escenarios de juego:

- Escenario 1: Es una cámara/habitación donde se encuentran varios guardianes enemigos, estos deben ser vencidos para que la puerta que se encuentra en la zona superior sea desbloqueada y pasar al escenario 2.
- Escenario 2: Es la cámara/habitación donde se encuentra el jefe Ganon.

Como punto extra se realizara un sistema de combate por turnos con el jefe Ganon que permita tener dos posibles ataques y la opción de no atacar o ceder el turno.

Esta implementación pretende ser realizada utilizando la librería OpenGL u otra similar y aplicando algunos de los conceptos estudiados a lo largo del curso.

Planteamiento

Para el desarrollo del proyecto se opta por utilizar el lenguaje de programación Python 2.7 y su librería gráfica PyGame la cual tiene por debajo OpenGL y muchas más utilidades construidas específicamente para facilitar la realización de video juegos en 2D. Debido a que es un juego 2D los aspectos de posicionamiento de cámara e iluminación se planearon para ser muy simples. La cámara siempre está sobre o arriba del escenario, como si se observara una caja sin techo y en su contenido los personajes. La transición entre escenas está planificada para no afectar el bucle principal del juego y este se realiza mediante el borrado de la escena y el dibujado de la siguiente escena. Además se pretende también crear otras pantallas tales como: escena de inicio con el título del juego y espera de evento por parte del jugador para empezar, de juego perdido y de juego ganado; estos últimos como posibles escenarios de finalización de la partida.

Se planeo haciendo uso de las comodidades que ofrece la librería la posibilidad de cargar imágenes, música y sonidos para ambientar y mejorar la experiencia de juego; para esto se consideró la búsqueda en internet de archivos de imágenes y sonidos característicos del icónico juego.

Comentarios

Uno de los conceptos vistos en la asignatura y que más se usó en la implementación del juego fue el de escalar figuras, en este caso imágenes. La librería **PyGame** maneja todo lo que está en escena como objetos **pygame.Surface** y **pygame.Rect**. Por ejemplo cuando cargamos imágenes estas generan un objeto **Surface** con las dimensiones de la imagen y muchas veces estas imágenes son muy grandes por lo que toca escalar estos objetos para ajustarse a nuestros requerimientos.

A pesar de que la especificación del proyecto no pide que los enemigos de la escena 1 puedan atacar; como iniciativa propia se planeo que si el jugador toca alguno de ellos, en consecuencia el personaje muere y el juego termina.

Consecuencias

Debido a que la librería PyGame no contiene un modulo para ajustar la posición de cámara ni para especificación de iluminación y debido a que ya se había empezado la implementación mediante la carga de imágenes en escena, el modo compartido con OpenGL no permite conjuntamente esas acciones. Así pues, no se aplica iluminación a las escenas por lo tanto las escenas solo contienen los colores de sus imágenes y elementos. Para el caso de la posición de cámara como se menciono anteriormente está fija, por lo tanto se crearon y diseñaron las imágenes de fondo de los escenarios mediante la ayuda de paint, con una especie de profundidad para dar la sensación en las paredes de las habitaciones de la escena.

Resultados

El proyecto finalmente queda compuesto por 9 archivos de código fuente:

- **director.py**
- **scene.py**
- **scene_home.py**
- **scene_stage1.py**
- **scene_ganon.py**
- **scene_battle.py**
- **scene_game_over.py**
- **scene_win.py**
- **main.py**

La clase **director.py** es la encargada de manejar todas las escenas del juego, esta clase ejecuta los métodos definidos dentro de cada clase escena del juego en el bucle principal y es el que hace posible la transición entre estas.

Para dar dinamismo al juego se crea la clase abstracta **scene.py** la cual recibe en su constructor un objeto director y contiene el esqueleto o interfaz que toda clase destinada a ser un escenario deberá heredar. Esta contiene los métodos que se usaran en el bucle principal del juego:

- **on_update():** para actualización lógica del juego
- **on_event():** se llama cuando llega un evento específico al bucle
- **on_draw(screen):** se llama para dibujar en pantalla, screen = pantalla o lienzo

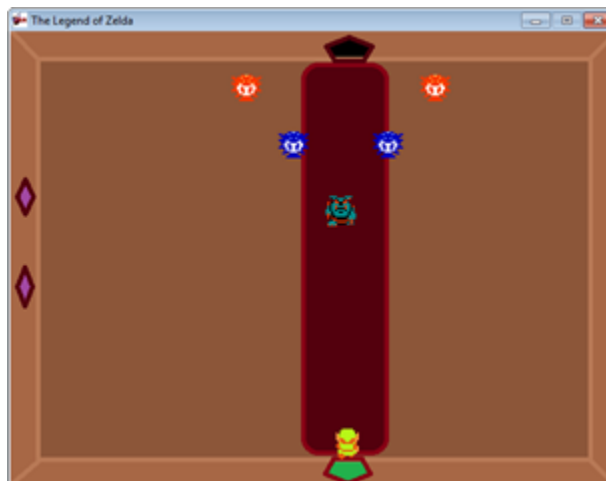
Las clases que heredan de la clase abstracta `scene.py` son las que se especifican en la forma: **scene_XXXXX.py** y representan cada escena del juego, estas contienen la lógica de cada una de ellas. El archivo **main.py** es como el nombre lo indica el programa principal que se debe ejecutar para dar inicio al juego.

A continuación se detallan los diferentes escenarios y pantallas del juego:

1) Pantalla introductoria con título del juego

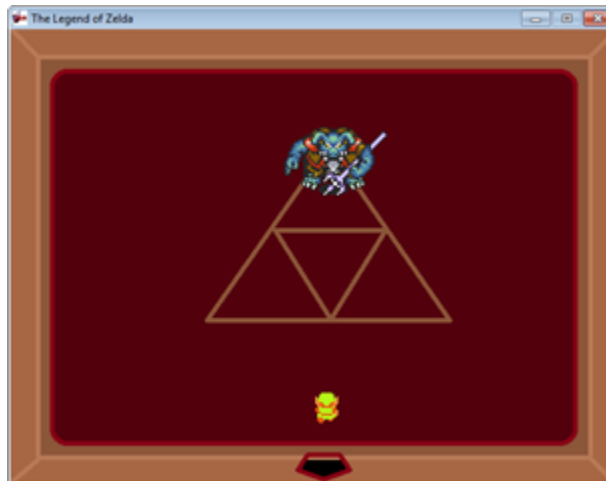


2) Escenario 1: Soldados enemigos.



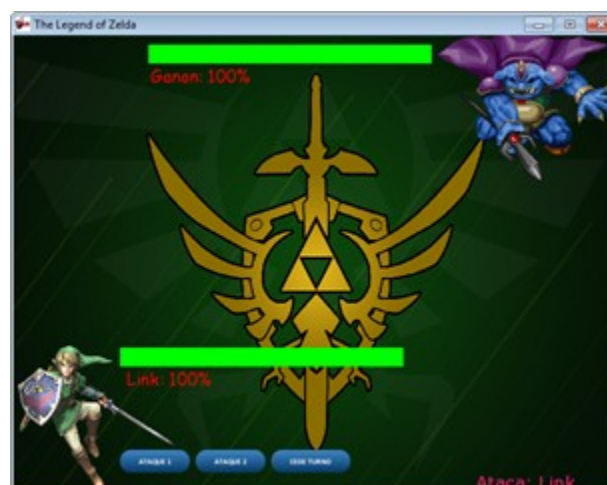
Link debe ser cuidadoso y no tocar ninguno de los villanos o morirá. Para matar los villanos se debe hacer click izquierdo sobre ellos. La puerta a Ganon será desbloqueada una vez todos sus soldados sean abatidos.

3) Escenario 2: Mazmorra de Ganon



Para enfrentarse a Ganon debes acercarte con Link y tocarlo o bien hacer click izquierdo sobre él.

4) Escenario de batalla. Sistema por turnos.



Este escenario presenta una interfaz para permitir atacar a Ganon con dos tipos de ataque diferentes y también la posibilidad de ceder el turno para atacar. Cada ataque produce un daño diferente. Lo mismo aplica para Ganon quien hace sus ataques aleatoriamente. La salud de cada uno de los contrincantes se representa en porcentaje para percibir la cantidad de daño infligido durante la contienda.

5) Pantalla de fin de juego cuando se vence a Ganon:



6) Pantalla de fin de juego cuando Ganon vence a Link o es tocado por alguno de los villanos de la escena 1:

