

Tarea 2 - Torre Helada

Profesora: María Cecilia Rivara Z.
Auxiliares: Sergio Leiva - Cristóbal Muñoz
Ayudantes: Darío Cáceres - Javier Díaz

1. Introducción

*IcyTower*¹ es un juego de plataformas que consiste en escalar una torre. La pantalla sigue al personaje constantemente, sin embargo, las plataformas desaparecen después de un tiempo. El juego termina cuando el jugador cae a un punto donde ya no quedan plataformas.



Figura 1: Screenshot del juego, dónde se ve al personaje subiendo por las plataformas.

2. Diseño de gráficos (3 pts.)

Para la primera parte de esta tarea, se le pide generar, utilizando Python, PyGame y PyOpenGL, Las siguiente componentes que seran parte del juego:

- **El personaje principal:** debe ser una figura compuesta por varias primitivas, por ejemplo, puede ser una persona o un animal. (0.6 pts.)
- **Los muros de la torre:** no pueden ser rectangulos planos aunque si puede ser un patron repetitivo y simetrico. (0.6 pts.)
- **Las plataformas:** debe haber por lo menos 3 diseños de plataformas, que no sean simplemente un rectangulo plano. Algunos ejemplos del juego son nubes, huesos o lianas. (0.6 pts.)

¹icytoweronline.hzik.com/

- **Un reloj:** Este llevara una cuenta regresiva de 30 segundos, su funcionalidad se explica más adelante. Como indicación, pueden modelar el marco del reloj y la manecilla por separado, y luego girar la manecilla utilizando las rotaciones de OpenGL. (0.6 pts.)
- **Escena inicial:** usando los diseños anteriores genere una escena estática del juego, debe tener por lo menos 5 plataformas. (0.6 pts.)

3. Mecánica Simple de Juego (1.5 pts.)

Se deben implementar las siguientes mecánicas de juego:

- El personaje debe chocar contra las murallas, y contra la parte superior de las plataformas solo cuando está cayendo (o sea que las puede traspasar cuando sube). Debe estar bajo algún tipo de gravedad. (0.4 pts.)
- El personaje se puede mover en 2 direcciones usando las teclas de flecha derecha y flecha izquierda, y puede saltar usando la tecla espacio. En este punto su velocidad y altura de salto son constantes. (0.4 pts.)
- La camara solo puede subir siguiendo al personaje, pero no baja. Si el personaje toca el fondo de la pantalla el juego termina. (0.4 pts.)
- Se debe tener una demo con por lo menos 50 plataformas para probar el juego, esto puede ser creado de manera estática y ser siempre igual. (0.3 pts.)

4. Características avanzadas (1.5 pts.)

Estas características son más difíciles de implementar que las anteriores. no es necesario mantener las funcionalidades anteriores al implementar estas.

- El movimiento del personaje hacia los lados debe ser acelerado, y la altura del salto depende de su velocidad actual. (0.5 pts.)
- Diseñar un "fin de juego" más complejo que el anterior, por ejemplo, en el juego original las plataformas desaparecen después de cierto tiempo y se pierde solo cuando se cae a un punto sin plataformas. (0.5 pts.)
- Las plataformas se deben generar con anchos y posiciones aleatorias. (0.5 pts.)

Los dibujos deben ser realizados con funciones de OpenGL en 2D y la ventana junto a las interacciones con el usuario mediante las funcionalidades de PyGame.

Bonus (máx 1 pto.)

- Agregar efectos de sonido.
- Utilizar sprites para el jugador y los elementos extra (el código para representarlos con primitivas más complejas igual debe existir).
- Implementar un sistema de puntaje que otorgue puntos a medida que el jugador sube la torre, y entregue más puntos por los saltos más altos (en terminos de plataformas).
- Agregar más diseños de personajes que se puedan elegir al inicio del juego, estos deben ser modelados con openGL y no ser simplemente sprites distintos.
- Agregar animaciones para el personaje cuando este salte y camine, como mover los brazos y girar en el aire.

Los bonus presentados son referenciales, si implementa más elementos extra, o añade más mecánicas propia de los juegos de este estilo, esto también será contabilizado (pero manteniéndose el máximo de un punto para bonus).

5. Presentación

Para la presentación se debe incluir:

- Breve descripción del problema.
- Esquema de algoritmos y métodos de resolución.
- Discusión sobre las dificultades encontradas.
- Resultados.
- Conclusión a partir de los resultados.

Entregar en formato PDF, de lo contrario no se revisará y se calificará con un 1.0

Condiciones de entrega

- La tarea es individual y las copias serán penalizadas.
- Entregas solo vía U-Cursos
- Adjuntar archivo Readme.txt con instrucciones de ejecución.
- Recuerde entregar la presentación en conjunto con la tarea. (En formato PDF)
- Recuerde adjuntar TODOS los archivos en cada entrega, ya que los auxiliares y ayudantes sólo tienen acceso a la última entrega de su tarea

Recomendaciones

- Revisar documentación y tutoriales de Python, PyGame y PyOpenGL para obtener en detalle las funciones disponibles y ejemplos de uso.
- Consultas a través del foro de U-Cursos o auxiliares.
- Sea ordenado con su código, agregue comentarios y utilice nombre útiles en sus variables y funciones.
- Planifique su tiempo y comience su tarea con anticipación. No comience a programar directamente. Comprenda el problema, realice esquemas y plantee un algoritmo de solución.