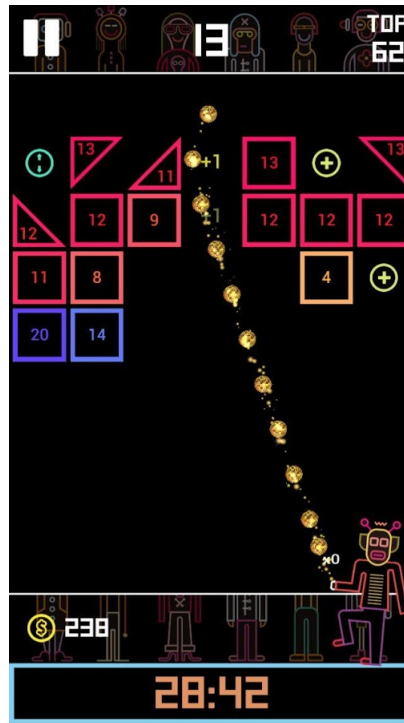


Tarea 2B

PP-Block

1. Introducción

La idea de esta tarea es diseñar e implementar un juego parecido a BB-Tan¹, donde el usuario dispara una serie de pelotas para destruir bloques que van bajando por turnos. Es una mezcla entre juegos tipo *Brick breaker* con *Bubble shooter*. Cuando algún bloque toca a la base, el jugador pierde.



¹Aquí pueden ver un gameplay del juego original: https://www.youtube.com/watch?v=242_2ynzDgs

2. Descripción de elementos

A continuación se definen algunos conceptos para mejor entendimiento de la tarea:

- a) **Pelota:** Objeto que será lanzado para destruir los bloques en pantalla. Puede tener diversas formas, tamaños y colores (con sus comportamientos respectivos). Al caer al borde inferior de la pantalla (excepto por la primera), esta se reúne con el jugador según las condiciones explicadas más adelante.
- b) **Bloque:** Elemento que será destruido por el jugador. Para destruir un bloque es necesario golpearlo tantas veces como su resistencia. La resistencia de los bloques es igual a la puntuación actual al momento de crearlo. En caso de que la puntuación corresponda a un número divisible por 10, la resistencia del bloque se duplica. Los bloques pueden ser cuadrados o triangulares.
- c) **Power-up:** Ícono ubicado en una celda que al contacto con una pelota producen algún efecto.
 - 1) **Extra-ball:** Aumenta en uno la cantidad de pelotas del jugador. Desaparece al contacto.
 - 2) **Vertical-laser:** Crea un laser de manera vertical, que disminuye en uno la resistencia de los bloques presentes en esa línea. Desaparece al caer la última pelota y solo si se produjo contacto con alguna de éstas.
 - 3) **Horizontal-laser:** Mismo comportamiento de la anterior, pero creando un laser de manera horizontal.
 - 4) **Change-path:** Cambia el ángulo de movimiento de la pelota de manera aleatoria al contacto. Desaparece al caer la última pelota y solo si se produjo contacto con alguna de éstas.
- d) **Filas y columnas:** La pantalla se divide en 9 filas y 7 columnas donde la fila superior siempre es vacía y la inferior debe estarlo (de lo contrario, el jugador pierde). Cada fila está compuesta por bloques y power-ups ubicados en las columnas. Por cada celda de la fila se dan las siguientes probabilidades:
 - 1) 50 % de ser un bloque.
 - 2) 25 % de ser un power-up.
 - 3) 25 % de ser vacío.

Una fila siempre debe contener al menos uno de cada caso nombrado anteriormente, donde el power-up corresponde a un extra-ball.

3. Descripción del juego

Inicialmente el usuario comienza con una pelota y centrado en la zona inferior de la pantalla. Al mismo tiempo, la puntuación parte en uno y aparecerá la primera fila de bloques a destruir.

El jugador determina el ángulo de trayectoria por el cual se lanzarán todas las pelotas continuamente hasta agotar su cantidad disponible. Cuando cae la primera, el jugador se cambia a esa posición y espera que caiga el resto de ellas. Al suceder esto último las filas actuales bajan a la siguiente, se incrementa en uno la puntuación, se rellena la segunda fila y se comprueba si la última fila contiene algo diferente a vacío. En caso de ser un power-up, desaparece; si es un bloque, el juego termina.

Para hacer el lanzamiento, el usuario debe hacer click con el mouse y arrastrar el puntero, creando una recta entre ambos puntos. Ésto define el ángulo y la velocidad del tiro.

Cuando una pelota queda en un ciclo de rebotes sin golpear bloques, se coloca un change-path en la columna central (en la misma fila donde se encuentra ésta) para cambiar la dirección de la pelota.

4. Implementación

Se le pide implementar un programa en Python, Pygame u PyOpenGL que cumpla con:

- a) Una versión simplificada del juego que incluya el comportamiento básico descrito anteriormente, vale decir, física de choques, acumulación de puntajes, lanzamiento de pelotas desde la columna central, generación y destrucción de bloques y power-ups **(3.0 pt)**.
- b) Un menú principal donde el usuario pueda seleccionar el tipo de pelota que quiere utilizar **(1.0 pt)**.
- c) Cambiar la posición de lanzamiento de pelotas en función de la primera pelota que caiga **(0.5 pt)**.
- d) Cambiar el color y agregar un número al bloque según su resistencia **(0.5 pt)**.
- e) Agregar el comportamiento de resistencia doble a los bloques bajo las condiciones explicadas anteriormente **(0.5pt)**.
- f) Agregar una guía o trayectoria de lanzamiento de pelotas **(0.5 pt)**.

5. Bonus

Además de los puntos anteriores usted puede hacer los siguientes bonus, **MAX 1pto**:

- a) Agregar nuevas formas de bloques y sus físicas **(0.5 pt)**.
- b) Agregar nuevos tipos de power-ups **(0.5 pt)**.
- c) Agregar una imagen que represente al jugador en la pantalla **(0.3 pt)**.
- d) Agregar música y efectos de sonido **(0.3 pt)**.

6. Informe

El informe debe incluir:

- Breve descripción del problema.
- Descripción de la solución.
- Discusión sobre las dificultades encontradas.
- Conclusiones.

Formato PDF: Los informes entregados en otro formato serán evaluados con nota 1.0

7. 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.

8. Recomendaciones

- Revisar documentación y tutoriales de Python para obtener en detalle las funciones disponibles y ejemplos de uso, en caso de programar con Python.
- Consultas a través del foro de U-Cursos.
- Sea ordenado con su código, agregue comentarios y utilice nombres 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.