PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 Programación Avanzada (II/2015)

Tarea 5

1. Objetivos

- Utilizar conocimientos de Threading para la creación de un programa.
- Desarrollar una interfaz gráfica utilizando PyQt.

2. Introducción

Debido a la infección zombie que se está esparciendo por todo el planeta, el gobierno necesita de un ejército que los pueda eliminar. Por esto, se ha decidido entrenar a las futuras generaciones para que ante una eventual invasión zombie la raza humana pueda sobrevivir. El problema es que los jóvenes solo están interesados en los videojuegos y no en luchar contra zombies. En respuesta a esto, se decidió tomar como primera medida la creación de un videojuego, en donde el usuario pueda enfrentarse contra zombies come-cerebros en un espacio cerrado. El gobierno le encargó esta tarea al Departamento de Computación de la Universidad, pero como usted bien sabe, para eso están los alumnos de Programación Avanzada.

3. Juego

El juego consiste en una persona controlada por el usuario mediante mouse y teclado, el cual se encuentra en un mapa rectangular cerrado (visto desde arriba) y sin obstáculos. Los zombies come-cerebros aparecen desde los bordes de la pantalla, dirigiéndose directamente al jugador con el fin de comer su cerebro. El jugador claramente no desea ser devorado, y además cuenta con una pistola, por lo que decide utilizarla para eliminar a los zombies. Lamentablemente la munición de su arma no es infinita y además pierde salud cada vez que hace contacto con un zombie. Para su fortuna, ha llegado a ayudarlo un helicóptero que lanza municiones y medicina al suelo cada cierta cantidad de tiempo (aún no se sabe por qué el helicóptero no decide simplemente rescatarlo).

3.1. Lógica del juego

Al correr el programa se deberá mostrar un menú que tenga al menos un botón para iniciar el juego. Además, mientras se juega, el usuario podrá pausar el juego ya sea apretando un botón en la ventana o bien presionando la tecla espacio. El juego debe poder reanudarse de la misma forma.

3.2. Personaje principal

3.2.1. Movimiento

El movimiento del personaje principal es controlado mendiante las flechas del teclado (o bien las teclas `WASD`) en conjunto con el puntero del mouse. El jugador siempre mirará hacia donde se encuentra el puntero del mouse y las teclas le permitirán moverse hacia adelánte, atrás, izquierda y derecha con respecto hacia donde esté mirando el personaje.

3.2.2. Disparos

Al hacer click el jugador disparará en dirección hacia el puntero del mouse. Cada disparo puede golpear a un único zombie, es decir que al colisionar con un enemigo, la bala deja de viajar. La trayectoria de la bala debe ser visible gráficamente. Además, el arma tiene un rango limitado, por lo que un disparo no viajará indefinidamente y eventualmente deberá desaparecer si es que no choca con nada. Cada disparo utiliza una munición y se debe mostrar en pantalla un contador con la munición restante.

3.2.3. Vida

Cada vez que un zombie haga contacto con el jugador le hace daño restándole salud. Se debe mostrar en pantalla de manera **gráfica** y **numérica** la salud del jugador.

3.3. Zombies

3.3.1. Llegadas

El tiempo entre llegadas distribuye Exponencial(λ). La tasa aumenta a medida que que avanza el tiempo de una partida, según una función $\lambda(t)$ definida por usted.

3.3.2. Movimiento

Los zombies siempre caminan en dirección al personaje principal. No obstante los zombies no pueden caminar por encima de otros. Debido a esto, ustede deberá implementar un sistema simple de colisiones que maneje esta situación. De igual forma, los zombies no podrán pasar por encima del jugador principal ni vice-versa.

3.3.3. Vida

Los zombies viven al límite en un estado que oscila entre la vida y la muerte. Por esto un simple disparo es suficiente para eliminarlos.

3.4. Ataque

Queda a su criterio la naturaleza de los ataques de los zombies, pero esta debe ser intuitiva y debe permitir un correcto desarrollo del juego. Un ataque no debe matar al jugador principal de un golpe, sin embargo debe hacer suficiente daño como para que el jugador tema por su vida. Se le recomienda considerar una demora entre los ataques de un mismo zombie, para que así el jugador pueda escapar si es que es atacado por un zombie. Los ataques deben ser mostrados de alguna forma gráfica para que así el jugador sepa que ha sido atacado.

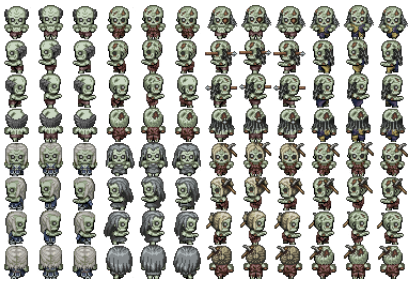
3.5. Helicóptero

Como se mencionó anteriormente, las municiones y la medicina se agotan, pero se pueden ir recuperando. El helicóptero cada cierto tiempo arroja municiones y medicina al campo de juego para suministrar al jugador. Estas aparecen al azar en cualquier lugar del mapa que se encuentre libre. No tiene por qué considerar colisiones con estos objetos. La cantidad de vida y municiones recuperadas queda criterio de suyo, al igual que la frecuencia con la que estas aparecen. Procure fijar una frecuencia que tenga sentido y que permita el correcto desarrollo del juego. **NOTA:** No es necesario que el helicoptero se vea gráficamente.

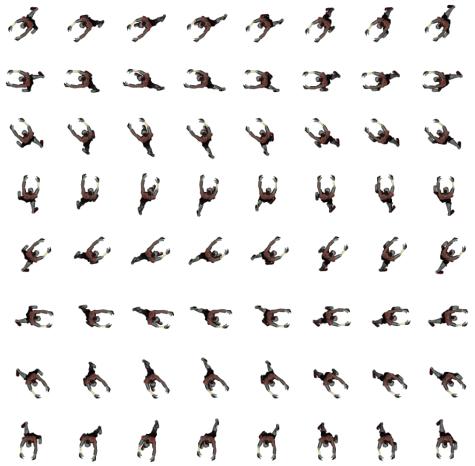
3.6. Representación Gráfica

Tanto los zombies como el personaje principal deben tener movimientos animados. Para lograr esto se debe diferenciar su aspecto al caminar en distintas direcciones. Además se deben observar al menos 3 estados de movimiento, por ejemplo, pies juntos, pie derecho adelante, pie izquierdo adelante, para que el movimiento sea aún más real. Si bien no es necesario que la muerte tenga una animación gráfica, se le pide que muestre de alguna forma clara que notifique cuando un zombies muera.

Para esta sección se le recomienda buscar en internet *sprites* de zombies y de shooters. Estas son imágenes que están pensadas para realizar juegos, las cuales muchas veces incluso ya incluyen las distintas orientaciones y estados al caminar/disparar/morir.



(a) Sprite Sheet vista diagonal.



(b) Sprite Sheet vista TopDown.

3.7. Puntaje

Durante cada partida se debe mostrar el puntaje del jugador, el cual se debe calcular mediante una función definida por usted que tome como parámetro el tiempo que ha transcurrido desde que partió el juego actual y la cantidad de zombies que ha matado. Se debe mostrar en tiempo real el puntaje que se va adquiriendo y al terminar la partida, el puntaje definitivo obtenido.

3.8. Notas

- Todas las magnitudes que no están especificadas en el enunciado quedan a su criterio. Lo importante es que las magnitudes elegidas aporten al correcto desarrollo del juego.
- Pueden basarse en el juego Age of Zombies a modo de ejemplo. Tome en consideración que este juego implementa muchas más cosas de las que se piden en esta tarea.

4. Restricciones y alcances

- Tu programa debe ser desarrollado en Python 3.4
- Esta tarea es estrictamente individual, y está regida por el Código de Honor de la Escuela: Clickear para Leer.
- Su código debe seguir la guía de estilos `PEP8`
- Si no se encuentra especificado en el enunciado, asuma que el uso de cualquier librería Python está prohibido. Pregunte por foro si se pueden usar librerías específicas.
- El ayudante puede castigar el puntaje[1] de tu tarea, si le parece adecuado. Se recomienda ordenar el código y ser lo más claro y eficiente posible en la creación algoritmos.
- Debe adjuntar un archivo `README.md` donde comente sus alcances y el funcionamiento de su sistema (*i.e.* manual de usuario) de forma *concisa* y *clara*.
- Cree un módulo para cada conjunto de clases. Divídalas por las relaciones y los tipos que poseen en común.
- Cualquier aspecto no especificado queda a su criterio, siempre que no pase por encima de ningún otro.

5. Entrega

- **Fecha/hora:** 9 de Noviembre - 23:59 hrs
- **Lugar:** GIT - Carpeta: Tareas/Tarea_05

Tareas que no cumplan con las restricciones señaladas en este enunciado tendrán la calificación mínima (1.0).

[1] Hasta −5 décimas.