



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

IIC2233 Programación Avanzada (I/2015)

Tarea 5

1. Objetivos

- Ampliar los conocimientos para la creación de interfaces gráficas.
- Aplicar los conceptos de Threading a un entorno de programación, específicamente un juego.
- Entender el funcionamiento de bytes y su aplicación para formar archivos.
- Usar serialización para almacenar estados del juego.

2. Problema

En esta tarea usted deberá programar un juego nunca antes visto, desarrollado por el DCC: *Poc-Mon*. La tarea consiste en crear la interfaz gráfica del juego junto con su lógica. La interfaz se deberá realizar utilizando PyQt. El DCC se inspiró (sólo un poco) en un conocido juego para desarrollar el suyo, por lo que se recomienda jugarlo para entender lo que se deberá programar. Lo puede encontrar en el siguiente enlace: **Enlace inofensivo** (Por ningún motivo esto se debe transformar en un acto de procrastinación).

3. Juego

3.1. Reglas

El objetivo del juego es recolectar todas las monedas (frutas/diamantes/lo que se le ocurra) del mapa para así avanzar al siguiente nivel. Por otro lado, en el mapa habrán otros personajes que le tienen mucho rencor al personaje principal y desean atraparlo. Debe tener cuidado con que otro personaje lo atrape, ya que en ese caso perderá una vida. Si el jugador pierde 3 vidas, perderá la partida y deberá comenzar de nuevo.

3.2. Personajes

Los personajes deben ser representados por imágenes. Estas imágenes se encuentran junto al enunciado. Cada personaje tiene una imagen representativa única.

Existen 5 tipos de personajes en el juego. El que usted controla es Poc-Mon y el resto son sus enemigos. El jugador debe escapar de sus enemigos y evitar que lo toquen. Cada entidad debe estar en constante movimiento. Cada enemigo tiene comportamientos y características diferentes. Es por esto que cada uno de ellos deberá ser un Thread diferente, que esté completamente desligado del resto.

Los enemigos reconocen paredes y al resto de las entidades en el juego. Los enemigos no se pueden atravesar entre sí. Los personajes del juego son:

- **Poc-Mon:** Es el personaje principal del juego y el único que usted puede controlar. Para controlar a este personaje se deben utilizar las flechas del teclado. La velocidad de Poc-Mon queda a su criterio (siempre y cuando sea posible jugar).
- **Mr Patiwi:** Es de color rojo y el líder de los enemigos. Mr Patiwi seguirá a Poc-Mon por todo el mapa e intentará atraparlo. Su velocidad es superior a la de Poc-Mon en un 20 por ciento.
- **Belenciwi:** Es de color rosado. Su objetivo es que Poc-Mon obtenga el menor puntaje posible, por lo que intentará recolectar la mayor cantidad de monedas que pueda. Este es el único personaje (Además de Poc-Mon) que puede recolectar monedas. La velocidad de Belenciwi es superior a la de Poc-Mon en un 10 por ciento.
- **Jaimiwi:** Es de color celeste. Este personaje tiene poderes sobrenaturales y es capaz de leer la mente de Poc-Mon, por lo que intentará intersectarlo en su ruta de escape. La velocidad de Jaimiwi es superior a la de Poc-Mon en un 10 por ciento.
- **Marquiwi:** Es de color naranja. Este personaje acaba de rendir una prueba de Programación Avanzada, por lo que recorre el mapa confundido y sin rumbo en específico. La velocidad de Marquiwi es superior a la de Poc-Mon en un 10 por ciento.

3.3. Monedas

Al comenzar el juego, cada casilla del mapa se encuentra con monedas. Existen diferentes monedas que usted, el jugador, puede recoger. Cada vez que el jugador recoge una moneda se generan distintas acciones en el juego. Las monedas del juego son:

- **Amarillas:** Son las más abundantes en el juego. Aparecen en el principio de cada nivel con una probabilidad de 0.9. Otorga 1 punto.
- **Rojas:** Aparecen al comienzo del juego con una probabilidad 0.10. Cuando Poc-Mon recoge esta moneda, se pone una máscara de "Período de pruebas", por lo que puede atrapar a sus enemigos y destruirlos (física y mentalmente) por un período de 10 segundos. Todos los enemigos cambian a color azul y se escapan desesperadamente de Poc-Mon de forma inteligente. Además, el jugador gana una vida. Otorga 10 puntos.
- **Verdes:** Aparecen de forma aleatoria en cualquier momento del juego, y solo 1 vez por nivel. Luego de recogerla, el comportamiento de los enemigos cambia al de uno de sus compañeros (incluido el color). Recordar que solo puede haber un enemigo de cada tipo en la partida, es decir, los enemigos rotan entre ellos. Otorga 30 puntos.
- **Azules:** Aparecen de forma aleatoria en cualquier momento del juego, y solo 1 vez por partida. Pueden aparecer en cualquier parte del mapa, incluso arriba de una moneda. Al recogerla se gana el juego de forma instantánea. Si después de 5 segundos no es capturada, esta desaparecerá.

3.4. Nivel de dificultad

Deberá programar para cada enemigo un comportamiento diferente, pero que tengan un nivel de dificultad en común. Deberá hacer este nivel de dificultad lo más difícil posible, indicando en el README todas sus consideraciones al crearlo (camino más eficiente hacia el objetivo, toma de decisiones, etc).

3.5. Mapas

En el mapa no pueden existir callejones sin salida. Las características del mapa quedan a su criterio, pero deberá especificarlo en el README.

Existen aspectos comunes que todo mapa debe cumplir:

- Forma: los mapas tienen la forma de un laberinto.
- Base: en el centro del mapa existe una base de donde emergen los enemigos.

3.6. Menú

Al comenzar el programa deberá mostrar en pantalla un menú. También se podrá acceder a este menú cuando se pause el programa. Este menú cuenta con 3 opciones:

1. Nueva partida: Se comienza una nueva partida
2. Guardar partida: Si es que se está jugando, se guarda la partida actual. Notar que al guardar, se debe almacenar el estado del juego tal como se encuentra en el momento de guardar.
3. Cargar partida: Aparecen las últimas partidas guardadas por el jugador. Al seleccionar alguna, se reanudará la partida desde donde se dejó pendiente.

3.7. Información del programa

3.7.1. Vidas

En cada nueva partida el jugador comienza con 3 vidas. Se pueden ganar vidas si es que recoge una moneda roja. El máximo de vidas que puede alcanzar es 5.

3.7.2. Puntaje

Se debe mostrar el puntaje de la partida actual. Además el máximo puntaje que se ha obtenido en todas las partidas que usted ha jugado. Al terminar la partida se debe poder ingresar el nombre del jugador. Se debe mostrar un ranking de todos los jugadores, y otro ranking con sus mejores partidas.

3.7.3. Tiempo

Mientras usted juega hay un cronómetro que muestra el tiempo actual de juego. Cada nivel puede durar como máximo 4 minutos. Si es que se sobrepasa este tiempo, se pierde la partida.

4. Estadísticas Finales

Luego de terminar una partida, después de ingresar su nombre para registrar su puntaje, se despliegan las estadísticas finales del juego. Se debe mostrar la siguiente información:

1. Mapa de calor: se debe mostrar una imagen generada por su programa que tenga un mapa de calor. Éste debe mostrar con un color más intenso los lugares más recurrentes por donde paso el jugador. Qué colores utilizar o características adicionales quedan a su criterio, pero deben estar especificadas en el README.
2. Tiempo en cada nivel: mostrar la cantidad de tiempo que se estuvo en cada nivel.

5. Bonus

A continuación se presentan 4 bonus, de los cuales puede seleccionar como máximo 2 para realizar. Debe indicar cuales realizó en el README, y todas las consideraciones que tomó para cada uno de ellos. Cada b́onus otorga un 5 % extra en su nota final.

- Modo editor de mapas: Debe crear una interfaz especializada para editar mapas. Específicamente se deben poder crear nuevos niveles. Las características a agregar para cada editor quedan a su criterio, pero se tendrá en cuenta la creatividad y que el editor sea intuitivo y fácil de usar.
- Modo editor de personajes: Debe crear una interfaz especializada para editar personajes. Específicamente se deben poder crear nuevos personajes con características personalizadas. Las características a agregar para cada editor quedan a su criterio, pero se tendrá en cuenta la creatividad y que el editor sea intuitivo y fácil de usar.
- Inteligencia Artificial conjunta: Para este bonus deberá lograr que los enemigos puedan coordinarse entre sí y ponerse de acuerdo para acorralar a Poc-Mon mientras cumplen sus objetivos individuales.
- Generación de mapas aleatorios. Para este bonus deberá crear un mapa aleatorio cada vez que comience un nivel. Todos los mapas generados deben seguir las reglas detalladas en los puntos anteriores.

6. Restricciones y alcances

- Tu programa debe ser desarrollado en Python 3.4
- 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 puntaje¹ de tu tarea, si le parece adecuado. Se recomienda ordenar el código y ser lo más claro y eficiente posible en la creación algoritmos.
- La revisión de la tarea será realizada con distintos archivos `.txt`.
- 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*.
- Cualquier aspecto no especificado queda a su criterio, siempre que no pase por encima de ningún otro.

7. Entrega

- **Fecha/hora:** Jueves 11 de Junio de 2015 / 23:59.
- **Lugar:** GIT - Carpeta: Tareas/Tarea_05

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

¹Hasta -5 décimas.