

### Informe proyecto 3 de Introducción a la Programación: Pythonia's Master

"Hmm ... No entiendo lo que dice tu código... Esto debería llevar comentarios de atributos. Y qué es esto de aquí?"

"Profe, self.timer es un contador que – "

"— Deberías usar más clases, el proyecto se enfoca mucho en eso. Quizá ayude."

"¡... Pero mis serpientes se tele-transportan! ¡Eso debería ser un bonus!"

Esas fueron las palabras de mi profesora exactamente ayer. Día en que pensaba que mi proyecto estaba casi terminado. Sólo me faltaban los efectos de los ítems y agregar las "decoraciones". Sin embargo, por alguna razón, con el premio de la "patita", mis serpientes se tele-transportaban al cuadro (0,0), y se quedaban ahí. Otras incluso reptaban hacia los muros más cercanos al (0,0) y se quedaban ahí. Fue ayer, a eso de las 18.00 horas, cuando decidí hacer un cambio drástico.

Transformé todo mi proyecto a clases. Muchas clases. Clases de ítems, de serpientes, de héroe, de muros, de puntos de dominación, incluso de contadores internos para el juego.

El cambio, si bien tomó mucho tiempo, resultó ser favorecedor. Era mucho más fácil trabajar con los personajes del juego, con cada objeto de la etapa, y el ítem "patita" por fin funcionó correctamente. La manera en que estaba haciendo ese premio específico antes era muy engorrosa y complicada, y, de alguna forma, había atributos (de las pocas clases que sí estaba usando), que se transformaban de listas a ints, de ints a listas, y así sucesivamente. Por eso, con el objeto serpiente, y usando atributos para juego que guardara listas de coordenadas en la forma [x,y], y atributos para juego que guardaran listas de objetos, fui capaz de darme los distintos casos para duplicar las serpientes. Fue así, como finalmente el ítem patita funcionó.

El otro gran problema que tuve con este proyecto, fue como esperar 3 segundos en un punto de dominación de una serpiente. En un principio, creía que podía tocar el punto de dominación, "marcarlo" de alguna forma y moverme libremente con el héroe, hasta que el contador marcara 3 segundos. Sin embargo, esto no tenía mucho sentido en el contexto del juego. Para resolver el problema de reiniciar el contador a 0 si el héroe no se quedaba en el punto de dominación de la serpiente, creé atributos internos al juego, self.contador\_heroe y self.contador\_serpiente, y, además, creé una clase "Timers", que ayudaba a identificar quién era el que había tocado el punto ya dominado. Con self.contador\_heroe y self.contador\_serpiente principalmente, logré contar los 3 segundos.

Es importante hacer notar, que, si se juega un rato con el juego, da la impresión que, cuando una serpiente se queda esperando en un punto de dominación verde, esta lo domina más rápido que cuando el héroe se queda esperando para dominar un punto de la serpiente. Puse prints en varias partes del código, sobre todo después de las partes en que se especifican estas condiciones en def despertar(self,segundos), y me percaté de que, en ambos casos, se cumplían los tres segundos cronológicos. El problema, pienso, es que la serpiente se queda "trabada" en el punto de dominación verde, y por cada segundo, su posición sigue aumentando, entonces cuando el atributo interno cuenta 3, la serpiente sale disparada. A pesar de esto, las reglas del juego se cumplen.

Hay ciertas cosas que quise implementar pero que, por alguna razón, no fue posible. Además, mi computador es del 2010 y ya no funciona como antes. Me fue difícil poder trabajar en el proyecto en ocasiones, ya que el programa se pegaba. Especialmente, no podía usar ningún while, no me era posible. Es por esto que en todo mi código aparecen for's, búsqueda en listas, listas multidimensionales, clases, muchos atributos y variables, pero no ciclos. De control de flujo, sólo pude usar ifs y elifs.

## Supuestos del juego

Mi idea del juego era respetar las reglas y seguirlas, intenté ser lo más detallista posible. Me fijé en cada detalle y opción o posibilidad que se me ocurriera. Estoy consciente que mi código es extenso, pero mi profesora me dijo que no me preocupara. Personalmente, creo que la extensión se debe en parte a que puse comentarios por todos lados. La razón por la que hice esto fue, justamente, por el diálogo que coloqué al principio de este informe. Pensé en la persona que fuera a corregir mi proyecto y me gustaría que se pudiese entender. Intenté ser lo más explicativa y detallista en las explicaciones y comentarios.

Hay algunos supuestos del juego que hice, pero que son más bien de diseño. Mi idea era hacer una historia con el juego, pero no fue posible. Lo que más quería era mostrar un mensaje al principio, en donde las serpientes, enfadadas, dijeran las reglas. Sin embargo, por más que intenté poner el comando de `self.escenario.muestra_mensaje(self,título,texto)` en distintas partes, nada funcionó. Lo puse en el inicio del `__init__` de la clase juego, al final, al inicio de `despierta`, al final de `despierta`, e incluso creé variables y atributos para reconocer si ya se había mostrado la ventana de bienvenida. El problema fue que, al intentar ejecutar el comando, mis serpientes y el jugador se movían, pero siempre de frente. Por eso, preferí poner un diálogo pequeño en la parte de abajo de la ventana del juego.

Otro supuesto importante del juego son las etapas. Personalmente, pienso que es monótono y aburrido tener que siempre comenzar en la misma etapa, y tener que pasar por varias para llegar a la etapa que se quiere. Por eso, en mi código creé una lista en el `__init__` de la clase juego con los nombres de los archivos de los tres mapas, y aleatoriamente se inicializa uno. La idea era, también, que fuera una “aventura” del héroe, que llegaba a las tierras de Pythonia y debía vencer a las serpientes. Una vez finalizado el juego entonces, el héroe se iría de las tierras a enseñar el lenguaje místico de Pythonia. Pensé incluso en usar el método `resetea()`, pero dada las líneas que llevaba en ese momento, pensé que el código se extendería demasiado. La única forma que se me ocurría, era copiando el `__init__` de la clase Juego en `despierta`.

Finalmente, en cuanto a ir más allá con el juego, creo que, esta vez, me ajusté estrictamente a lo necesario. No hice los bonus propuestos por una cuestión de tiempo, y quise enfocarme en que lo que hiciera estuviese bien hecho. Sin embargo, si es cierto que el juego que hice tiene un cierto toque, en el sentido de crear el ambiente de una historia; un ambiente de personajes, en donde dialogan y tienen comentarios. Sobre todo, el final del héroe, si lo asfixian las serpientes, es un poco triste, pero a la vez cómico. Por tanto, considero lo anterior solo como un “extra” en cuanto a estilo y diseño, pero no hay mucho que vaya más allá en cuanto a programación se refiere.