Instituto Tecnológico de Costa Rica

Ingeniería en Computadores

Jeff Schmidt

Proyecto PacMan

Alexis Gavriel Gómez

2016085662

Contents

[Introducción 3](#_Toc467183840)

[Descripción del problema 3](#_Toc467183841)

[Diagrama de clases 3](#_Toc467183842)

[Análisis de resultados 3](#_Toc467183843)

[Dificultades encontradas 4](#_Toc467183844)

[Bitácora de actividades 5](#_Toc467183845)

[Estadística de tiempos 7](#_Toc467183846)

[Conclusión personal 7](#_Toc467183847)

# Introducción

El presente trabajo tiene como objetivo el aprendizaje de la interfaz gráfica en el lenguaje Python, utilización de objetos con atributos y métodos y manejo de archivos. Mediante la programación de un videojuego asimilado al videojuego PacMan. En este proyecto se programó en Python 3.5.1 utilizando la librería Pygame, también se utilizaron otras librerías como random, thread, mixer y time. El programa contiene cuatro ventanas, la inicial donde se ingresa el nombre del jugador, la ventana de ayuda, la ventana de puntuaciones más altas y la ventana del juego.

# Descripción del problema

El juego consiste en obtener puntos moviendo el jugador recolectando comida distribuida en mapa mientras se evita colisionar con los fantasmas. Si se consigue una capsula durante los próximos cinco segundos el jugador si colisiona con un fantasma hará que el fantasma regrese al inicio y obtendrá puntos extra. Si se consigue recolectar toda la comida del mapa y todas las cápsulas se incrementará el nivel de dificultad aumentando la velocidad de los fantasmas.

# Diagrama de clases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Juego |  | PacMan |  | Fantasma |
| Numero de Juego  Matriz  Comida  Capsulas  Scores | Estado  Posición\_x  Posición\_y  velocidad | Estado  Posición\_x  Posición\_y  Velocidad  Color |
| Comida  Capsulas | Get.color  Mov\_izq  Mov\_der  Mov\_arriba  Mov\_abajo | Mov  Mov\_izq  Mov\_der  Mov\_arriba  Mov\_abajo |

# Análisis de resultados

Utilizar una matriz que contiene las posiciones y los datos en cada posición ayuda a simplificar la codificación de colisiones ya que estas ocurren si dos objetos están en la misma posición en matriz.

El análisis de eventos en la ventana se puede clasificar en cierre de ventana, eventos de ingreso, como clics e ingreso de teclas, estos eventos permiten la toma de datos del jugador para que este se pueda mover en el juego.

La utilización de clases permite categorizar datos que pertenecen a un objeto, utilizar atributos y métodos permite utilizar los datos pertenecientes a este objeto para realizar varias funciones sobre estos

La utilización de variables globales ayuda a facilitar el uso de datos dentro y fuera de funciones

La iteración, junto con temporizador son muy importantes a la hora de dibujar en una ventana.

Las funciones de ordenamiento de matrices innatas de Python ayudan a reducir la complejidad de ordenamiento de datos.

El uso de un contador para reducir la frecuencia de la realización de un movimiento es una forma de aumentar la velocidad de movimiento.

El uso de variables temporales es importante para poder utilizar un temporizador para realizar actividades después de cierto lapso de tiempo.

La utilización de librerías que leen archivos de audio en formato mp3 reduce el tamaño requerido de los archivos de audio

# Dificultades encontradas

Movimiento del jugador sin que este atraviese las paredes. El error se encontraba en el índice a buscar, ya que la primera posición de coordenada dentro de una matriz está dada por fila, columna. Mientras que las posiciones de un dibujo están dadas, o sea: largo x ancho, columna x fila.

Guardar datos para que estos se utilicen desde otras funciones con variables globales, se hubiera preferido crear objetos globales y luego modificar estos.

# Bitácora de actividades

**29 octubre**

2 horas

Se instaló pygame para python 3.5.X

Se buscaron ejemplos de interfaz en pygame

Se crearon botones y dos ventanas

**1 noviembre**

2 horas

Se empezó la ventana de juego, y se creó el objeto pacman, el cual se puede desplazar sobre la pantalla sin que este se salga de ella.

Se decide dibujar la matriz de juego leyendo un archivo en lugar de que esta sea una imagen importada.

Se crea el archivo de la matriz

**2 de noviembre**

1 hora

Se logra crear la matriz de juego leyendo el archivo.

**5 noviembre**

3 horas

Se crean las funciones de movimiento para el jugador de manera que este no atraviese las paredes. Se crea un debbuger para conocer los alrededores del jugador (alrededores a su posición en la matriz), se tiene problemas con el movimiento ya que no se logra ubicar correctamente las paredes

**6 noviembre**

4 horas

Se logra corregir los errores del movimiento, el jugador se mueve correctamente en el mapa de juego. Se crean los fantasmas que se mueven por el mapa de manera aleatoria, si se colisiona con estos el juego finaliza. Se cambian los datos en la matriz para que estos contengan las posiciones de la comida, capsulas, camino libre y paredes. Se consigue dibujar en la matriz estos objetos. Se logra “comer” la comida, obtener puntos y desaparecer el objeto del mapa.

**7 noviembre**

1 hora

Se buscan formas para mover los fantasmas y jugador a diferentes velocidades

**9 noviembre**

Se logra crear el estado del jugador en donde este puede comer a los fantasmas, esto se acciona con una tecla y se finaliza con una tecla. Se crea un contador de tiempo para que demore en este estado 5 segundos, y se regrese al estado anterior automáticamente. Durante este estado la velocidad del jugador aumenta ya que se desplaza más distancia por ciclo.

**12 noviembre**

5 horas

Se decide utilizar un contador de tiempo (1/100 de segundo) para llevar el control de las velocidades de los fantasmas y jugador. Se adapta la parte del código del dibujo en la pantalla. Se cambia la velocidad de los fantasmas para que el fantasma rojo sea más rápido. De manera similar se aumenta la velocidad del jugador al cambiar de estado al condicionar la toma de ingreso de los comandos de movimiento utilizando el contador.

**13 noviembre**

4 horas

Se agregan condiciones para las puntuaciones. Se agregan las capsulas en el mapa, las cuales cambian el estado del jugador. Se crean multiniveles al comer toda la comida y capsulas, el multinivel aumenta la velocidad de los fantasmas. Se tienen problemas con el movimiento de los fantasmas al intentar modificar paredes del mapa para que estos sean los únicos que puedan atravesar.

**15 noviembre**

2 horas

Se crea la ventana de puntuaciones, en esta se despliegan los datos de una matriz de puntuaciones. Estas son modificadas y leídas de un archivo de texto para que contenga las puntuaciones máximas actualizadas.

**16 noviembre**

2 horas

Se crea la ventana de ayuda y se modifica la interfaz de las ventanas: ayuda, puntuaciones e inicio, para que estas sean más agradables y entendibles, se utilizan imágenes como fondo en las ventanas. Se utiliza un hilo para reproducir sonido en la ventana de inicio

**17 noviembre**

4 horas

Se creó la documentación externa, e interna.

# 

# Estadística de tiempos

|  |  |
| --- | --- |
| Función | horas |
| Análisis de requerimientos | 2 |
| Diseño de la aplicación | 2 |
| Investigación de funciones | 5 |
| Programación | 15 |
| Documentación interna | 2 |
| Pruebas | 2 |
| Elaboración documento | 2 |
| Total | 30 |

# Conclusión personal

La creación de un videojuego conlleva un gran número de variables, técnicas, funciones lógicas a aprender. Además de que un videojuego requiere de muchas variables siendo controladas en el mismo tiempo.