

Taller de programación

Proyecto programado #1

Profesor: Antonio Gonzalez Torres

Estudiante: Adrián López Vásquez

Carné: 2016108981

Grupo 04

Fecha entrega: 15 de mayo de 2018

I Semestre 2018

Índice:

1) Introducción.....	3
2) Descripción del problema.....	4
3) Análisis de resultados	5
4) Bitácora de actividades	10
5) Estadística de tiempos	12
6) Conclusiones	13
7) Bibliografía	14

1) Introducción:

La programación trae consigo una serie de desafíos que permiten que el programador aprenda a desarrollar habilidades de programación. Es por ello que mediante este juego se desarrollaron una serie de conocimientos extraclase que permiten familiarizarse con el lenguaje de programación Python, la utilización de clases, el uso de json para el almacenamiento de variables. Con este proyecto se permite que mediante un juego se evalúen los conocimientos que se tienen sobre la programación.

PyDeathRace es un juego en el que, mediante un circuito, se da la competencia entre dos jugadores en 3 minutos usando carros controlados por el teclado de la computadora. Este juego permite conocer más a fondo las funciones de pygame, la programación orientada a objetos y el uso de sprites y superficies.

2) Descripción del problema

El juego como tal debe seguir una serie de reglas definidas anteriormente. PyDeathRace es un juego basado en los juegos de carros antiguos programado usando Python y pygame , para el desarrollo del juego se debe contar con un escenario tipo pista en el cual el vehículo al pasar por una sección(meta) contabilice las vueltas que ha dado el vehículo. Se debe contar con una forma en que dos jugadores jueguen simultáneamente y disparen. Al girar el vehículo se debe actualizar este permitiendo que el vehículo se desplace en la dirección rotada

Al analizar el problema se tienen varios desafíos para el desarrollo del juego. El vehículo debe desplazarse usando el teclado. Se debe contar con un limite que permita que el vehículo no sobrepase los limites de la pista que se toma que se baje la aceleración del carro y así este pierda tiempo desplazándose.

Para guardar las mejores puntuaciones se debe hacer uso del json y este debe poder modificarse adrede para poder actualizarse con cada partida. Se debe controlar una serie de dummy vehicles mediante la utilización de clases y además se tiene que contar con un menú que permita la selección entre puntaje, jugar y salir.

3) Análisis de resultados

Primeramente, se realiza una búsqueda de las distintas funciones de Pygame en su propia documentación. En ella se encuentra el uso de sprites y ejemplos de como utilizar pygame. Toda la documentación buscada se encuentra en la bibliografía.

Se realizó una función para que el carro avanzara posteriormente y se definió un fondo usando rectángulos. Para el avance del carro se logro que este girara usando la proyección de vectores cambiando de coordenadas rectangulares x y y a utilizar el seno y coseno buscando un angulo que relacionara ambas coordenadas. Como primera fase el proyecto lucia de la siguiente manera:

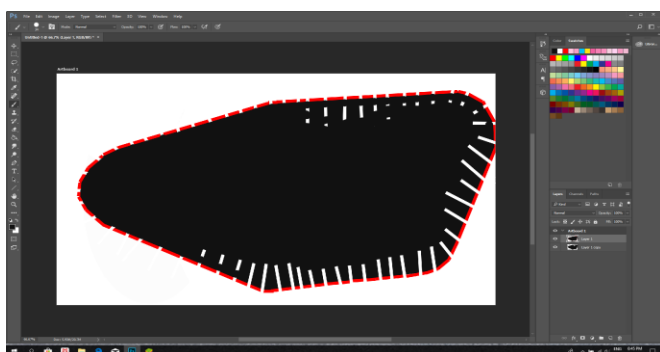


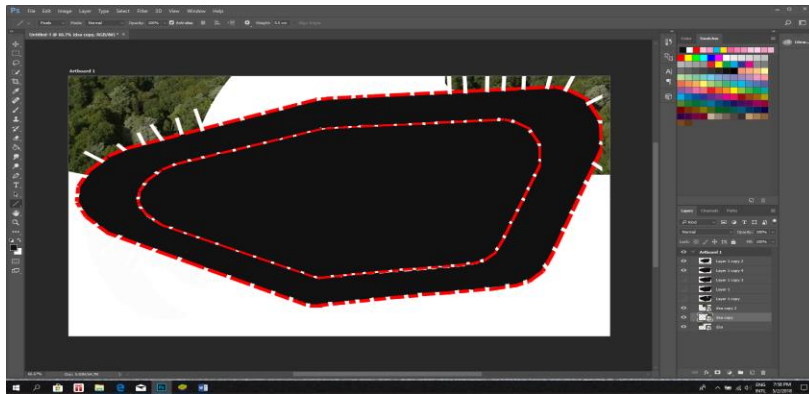
Se quería basar el escenario en una pista de carreras para ellos se uso el programa Photoshop para realizar el escenario.

Se utilizaron las siguientes imágenes para hacer el fondo:



Se transformo la pista hasta llegar al resultado obtenido





Al terminar la pista se aplico a la pantalla y se empezó a desarrollar una forma de que el vehículo disparara. Se implementaron las clases y se empezó a diseñar el menú en Photoshop.



Por ultimo se realizo los métodos para guardar las puntuaciones en json, implementación del modo para dos jugadores, aceptar los nombres de los jugadores y ver los puntajes.

Al finalizar el código el proyecto tomó la siguiente forma:





Se escogió que los límites de la pista se determinaran por el color así si se dale de esto la aceleración es baja. Se logró la implementación de json a Python como modo para cargar un archivo y editarlo. Para mejorar el programa se deben considerar mejor los límites de la pista y también se debe enfocar en los dummy ya que no se logro la completa independencia de estos debido a la desuniformidad del circuito. Como se ve en las imágenes se cumplió con la mayoría de requerimientos solicitados y se amplió el conocimiento en pygame, sprites, uso de json, y programación usando clases. Por su parte se tiene un juego que corre bastante agil. El mayor problema se presenta en los dummy ya que para realizarlos se debe tener mucha precisión al seleccionar los puntos porque si no, no llegan a la misma posición y se desfasan. El juego corre oportunamente para dos jugadores y se lograron los cambios de aceleración.

4) Bitácora de actividades

Análisis del problema: Se investigó sobre pygame y Tkinter y se analizó el problema y sus dificultades, se optó empezar por la movilidad. Duración 1 hora

Investigación: Se investigó la documentación de pygame, se estudiaron ejemplos, tutoriales, se redondeó en el uso de sprites y la colocación de imágenes en el juego. Duración=4 horas

Desarrollo movilidad carro: Se ideó una forma de movilidad del carro usando la proyección de vectores mediante cosenos y senos y se investigó si existían formas más eficientes para desplazar el carro cosa que no se obtuvo. Duración: 2 horas

Creación fondo pista: Se creó usando Photoshop la imagen usada como fondo de la pista usando una serie de imágenes, se basó en una pista de carreras. Duración 15 min

Implementación de clases: El código que se tenía se transformó usando clases y métodos para ello se investigó sobre las clases en Python. Duración 1 hora

Realización del menú: Se realizó un nuevo proyecto y se hicieron los botones y controles del menú, posteriormente estos fueron trasladados al código principal. Duración 30 min

Fondo menú: Mediante el uso de Photoshop se creo la imagen del menú. Duración 15 min.

Disparos: Se realizó una clase y métodos que permitieron al jugador disparar una bala en la posición en que este se traslada. Duración 1 hora

Modo 2 jugadores: Se utilizó el código existente y se creó un nuevo objeto que permitió que dos jugadores puedan al mismo tiempo jugar. Duración 30 min

Límites de la pista: Se ideo una manera de que la pista tuviera límites y fueran respetados por los jugadores por ello se utilizó el color negro de la pista como método de validación. Duración 3 horas

PedirNombrejugador: Se desarrollo una forma en la que mediante un rectángulo se procesa texto y se obtiene el nombre del jugador para posteriormente usarlo en el código. Duración 30 min

Implementación Json: Una parte complicada fue el uso de json ya que se tuvo que investigar bastante sobre el uso de este y se realizó un método que abre el json y saca los primeros, segundos y terceros lugares. Duración 1 hora

Creación de ventana puntuaciones y ventana pedir nombre: Se implemento texto y un fondo en cada una de estas ventas para seguir con el estilo del juego. Duración 20 min

Mejoras código: Se realizaron una serie de mejoras al código acortando pasos y usando menos variables. Duración 30 min

Documentación interna: Se comento cada una de las funciones en el código y que realiza el código en cada sección y cual era la intención buscada. Duración 1 hora

Elaboración documento: Se realizó un análisis de lo realizado en todo el proyecto y se recopiló en este documento. Duración 1 hora y media.

5) Estadística de tiempos

Tabla 1. Tiempo utilizado en el desarrollo de las actividades

Actividad	Tiempo
Análisis del problema	1 hora
Investigación	1 horas
Desarrollo movilidad carro	2 horas
Creación fondo pista	15 min
Implementación de clases	1 hora
Realización del menú	30 min
Fondo menú	15 min

Disparos	1 hora
Modo 2 jugadores	30 min
Límites de la pista	1 horas
Pedir nombre jugador	30 min
Implementación Json	1 hora
Creación de ventana puntuaciones y ventana pedir nombre:	20 min
Mejoras código	30 min
Documentación interna	1 hora
Elaboración documento	1 hora y media
Total	12 horas 20 min

6) Conclusiones

Como se pudo ver se puede realizar un juego utilizando Python como lenguaje de programación. El juego requiere de una gran cantidad de tiempo como se puede ver en el diagrama ya que requiere de investigación y adaptación del problema. No se pudo realizar el control automático de los dummy ya que el escenario es un poco irregular lo cual dificulta esto. Sin embargo, se realizaron las demás partes requeridas del programa. Por ello se ve que el juego en sí funciona y cumple con la mayoría de requerimientos.

7) Bibliografía

(s.f.). Recuperado el 11 de Mayo de 2018, de <https://pythonprogramming.net/making-interactive-pygame-buttons/>

(s.f.). Recuperado el 7 de Mayo de 2018, de <http://www.pygame.org/docs/ref/transform.html>

Craven, P. V. (s.f.). Recuperado el 10 de Mayo de 2018, de http://programarcadegames.com/index.php?chapter=introduction_to_sprites&lang=en

Jens, H. (s.f.). Recuperado el 10 de Mayo de 2018, de <http://thepythongamebook.com/start>

realpython. (s.f.). Recuperado el 10 de Mayo de 2018, de <https://realpython.com/python-json/>

Robinson, S. (17 de Agosto de 2016). *stackabuse*. Obtenido de Reading and Writing JSON to a File in Python: <http://stackabuse.com>

Sridhar, J. (11 de Agosto de 2017). *Makeuseof*. Obtenido de <https://www.makeuseof.com/tag/json-python-parsing-simple-guide/>