Documentación Externa

Proyecto #2

Taller de Programación Gr4

Mario Carranza Castillo

2019180212

Álvaro González Ballesteros

2019379863

TEC | Tecnológico de Costa Rica

Tabla de Contenidos

Introducción

Con el presente proyecto se pretende la creación de un juego de carreras en 2D, Para efectos de su realización, se estarán utilizando los programas, Tkinter y Pygame destinados en gran parte para tal fin.

Se pretende, con el trabajo en este video juego, adquirir, por medio de investigaciones a fondo, los conocimientos necesarios para poder trabajar con los programas TKinter y Pygame en la creación y ejecución de video juegos en 2D.

Durante el trabajo se reqirió de la inversión de mucho tiempo y dedicación, aunque inicialmente se realizó una amplia investigación sobre la forma en que ambos programas trabajan, por ejemplo, el lenguaje que utilizan y los diferentes algoritmos que pueden usar para cargar imágenes, sonidos, etc. ya ambos compañeros teníamos previo conocimiento en estos programas, de igual manera al requerir el juego muchas funciones diferentes al las ya estudiadas, como la creación de un servidor para lograr que un jugador cualquiera pueda jugar contra otro en un dispositivo diferente, nos vimos en la completa necesidad de ampliar la imformacion que se tenia en cuanto al uso de dichas fuentes de programación.

Para efectos de este proyecto se implementó pygame para la escritura del código del juego debido a que tiene funciones con las que se nos hizo más fácil trabajar en cuanto a lógica, y además se implementó también el uso de tkinter para el diseño de la pantalla principal del juego ya que se nos hizo más fácil para establecer diferentes botones con diferentes funciones que nos ayudarían a ingresar a los datos necesarios e incluso al juego en sí. Para todo ello se unieron ambos codigos (el que se hizo en tkinter y los realizados en pygame) con el fin de obtener una interfaz de juego agradable.

se aprende gran cantidad de cosas nuevas que servirán de mucho en proyectos futuros.

Con todo lo realizado durante este proyecto se ve la gran posibilidad de crear juegos a partir de programas con lenguajes sencillos, como lo son Pygame y Tkinter, y que aunque no tienen las funciones necesarias para hacer trabajos más elaborados

como juegos en 3D, son una muy buena forma de empezar a entender y adentrarse en el mundo se los video juegos.

Descripción del Problema

Programar una versión del juego DakarDeath en Python, TKinter y Pygame (combinados) utilizando una arquitectura cliente-servidor en el que pueden competir dos o más jugadores, y cada jugador usa una computadora diferente. El juego utilizará una arquitectura cliente servidor, en la cual el tablero es controlado por el servidor y desplegado por cada cliente en su computadora. Las acciones que realizan los clientes son enviadas al servidor y este actualiza los tableros de todos los jugadores participantes. El formato de la pista y el diseño de los carros lo debe definir cada grupo.

La velocidad y dirección de cada automóvil es controlada por el jugador con las teclas de su computadora. El juego contará con al menos tres niveles de dificultad y su menú tendrá las siguientes opciones:

1. Registrar nombre de usuario
2. Ver puntaje
3. Iniciar partida
4. Guardar partida
5. Reiniciar partida
6. Salir

Los jugadores pasan de nivel cuando han superado 2 minutos de juego sin sufrir una colisión. Los niveles superiores del juego cuentan con más obstáculos (definidos por el programador, la mayor dificultad favorece el otorgamiento de más puntos), como mayor número de carros, rocas, muros y árboles. Además, el juego cuenta con sonidos de aceleración, disminución de la velocidad, frenado y explosiones al chocar.

En este juego el competidor debe sobrepasar o eliminar a los oponentes que son controlados por el mismo juego. Las reglas básicas del juego son las siguientes:

1. El juego se desarrolla en un desierto lleno de minas y obstáculos y no existe una pista de carreras, por lo que los vehículos pueden correr de forma libre por cualquier parte.

2. El tiempo de cada partida del juego es de 2 minutos.

3. El juego permite que dos o más jugadores compitan a la vez.

4. El sistema controla un gran número de “dummy vehicles” que corren a una velocidad constante.

5. El punto de partida de la competencia y llegada se encuentra indicado por una zona marcada como un tablero de cuadrículas.

6. La velocidad de los vehículos de los jugadores es variable y depende de las maniobras que hagan con las teclas de las computadoras.

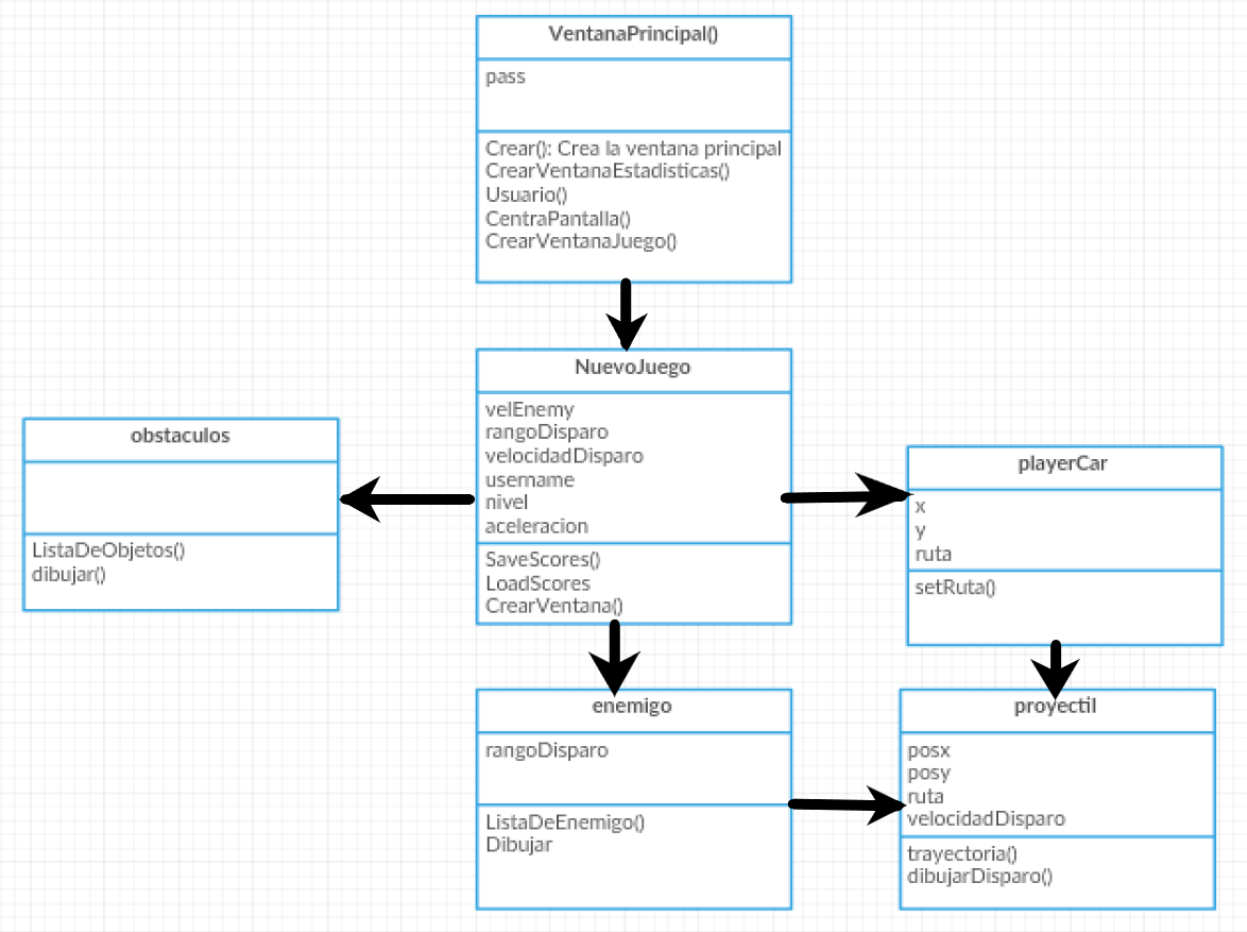
7. El puntaje de cada jugador se calcula con base en el número de “dummy vehicles” que elimina durante la competencia y la distancia recorrida.

8. Un jugador puede eliminar el vehículo del competidor rival y los “dummy vehicles” empujándolo para que se destruya contra un obstáculo o caiga a un barranco y también mediante disparos (puede usar cualquier método de destrucción).

9. El jugador que se sale de la pista y no se destruye sufre el rebajo de puntos, de acuerdo con las reglas que defina el programador.

10. Si un jugador sufre una colisión con un objeto letal al salirse de la pista puede quedar destruido.

Diagrama UML



Análisis de Resultados

Llegando a la conclusión del proyecto, aprendimos el manejo de sockets para servidores y clientes y asi poder mostrar información en varias computadoras, guardando todos los datos en archivos con pickle. También con este mismo manejo de archivos, poder guardar y reiniciar una partida guardada.

Asi mismo reforzamos los conocimientos adquiridos en el proyecto anterior los cuales eran manejo de las librerías Tkinter para hacer el menú principal con las opciones de ver las estadísticas, crear usuario, etc. Usando objetos tales como listboxes, textboxes, botones, etiquetas, entre otros. Y también de la librería Pygame para hacer todo el entorno del juego.

Reforzamos el método de manejo de archivo para guardar puntuaciones con csv y obtener los mejores resultados con JSON.

Aprendimos también como simular una ventana infinita, haciendo el fondo móvil, poner obstáculos con posiciones aleatorias en todo el mapa y el movimiento aleatorios de los bots o dummy vehicles.

Bitácora de actividades

Lunes 6 de mayo:

-Se realizó una pequeña búsqueda de los componentes del juego (imágenes, sonidos, etc), se buscó información sobre los “sockets” (como funcionan y como usarlos), y se creó una pantalla destinada a ser la pantalla principal del juego.

Duración aproximada de la actividad: 2.5 horas.

Miércoles 8 de mayo:

-Se terminó de buscar las imágenes para el juego, se buscó más información sobre los “sockets” y con ello, se hicieron los usuarios y el servidor, se trabajó en el código principal del juego como por ejemplo la creación de la clase principal, la clase para el jugador con sus respectivas funciones, la clase de los enemigos (no se terminó).

Duración aproximada de la actividad: 6 horas.

Jueves 9 de mayo:

-Se trabajó más en la creación del servidor para el juego (búsqueda de información y creación del servidor para que el juego se vea en las 2 computadoras).

-Se buscó información para hacer que se mueva el mapa del juego y crear una sensación de que el carro de los jugadores.

Duración aproximada de la actividad: 6 horas.

Sábado 11 de mayo:

-Se hizo la validación correspondiente para el movimiento de mapa según la tecla de navegación presionada.

Duración aproximada de la actividad: 3 horas.

Domingo 12 de mayo:

-Se realizó una búsqueda de imágenes para obstáculos, línea de meta.

Duración aproximada de la actividad: 1 hora.

Lunes 13 de mayo:

-Creación de clase para los enemigos controlados por el sistema.

-Creación de clase para los obstáculos utilizados en el juego.

Duración aproximada de la actividad: 5 horas.

Martes 14 de mayo:

-Se añadió el reloj para controlar niveles.

-Duración aproximada de la actividad: 2 horas.

Miércoles 15 de mayo:

-Se añadió barra de estado para el control de la vida del jugador.

-Se añadió dentro de los obstáculos una imagen que recupera una parte de la vida del jugador.

Duración aproximada de la actividad: 3 horas.

Jueves 16 de mayo:

-Se agrego aceleración progresiva al jugador.

-Se validó que los enemigos se destruyan al dispararles.

Duración aproximada de la actividad: 4 horas.

Lunes 20 de mayo:

-Se hizo el código para guardar, reiniciar y salir de la partida.

-Se pusieron minas dentro de la lista de obstáculos para que al colisionar con una de ellas la vida del jugador se reduzca considerablemente.

Duración aproximada de la actividad: 6 horas

Martes 21 de mayo:

-Se hizo el código para que el jugador pueda empujar a los enemigos contra obstáculos.

-Se validaron las puntuaciones recibidas y se guardaron en los archivos.

-Se puso la bandera de meta.

-Se validaron los niveles a partir de cada dos minutos de sobrevivencia.

-El jugador disminuye la velocidad al dejar de presionar las teclas de movimiento.

Duración aproximada de la actividad: 6 horas.

Sábado 25 de mayo:

-Se actualizaron los efectos de sonido y se validaron según se requirió.

-Se validaron las posiciones de todas las imágenes dentro del servidor.

-Se validó la acción a realizar si el jugador colisiona o sobrepasa la línea de meta, en este caso se pasa al siguiente nivel.

Duración aproximada de la actividad: 5 horas.

Estadísticas de Tiempos

Análisis de requerimientos	1 horas
Diseño de la aplicación y diagrama de clases	22 horas
Investigación de funciones	3 horas
Programación	30 horas
Documentación interna	2 horas
Pruebas	22 horas
Elaboración documento	4 horas
TOTAL	84 horas

Conclusiones

- Se logro terminar el proyecto según los parámetros indicados.
- Se reflejaron las horas trabajadas en un cuadro y en la bitácora.
- Se logro el aprendizaje de los módulos socket y pickle para servidor y cliente