

Proyecto Programado #4

luiKi Kart

Introducción

Los juegos de carreras han sido desarrollados y distribuidos por diferentes empresas, y estos cuentan con la aparición de personajes, que pueden ser adaptados a los gustos de los jugadores. A lo largo de la historia de los video juegos, los juegos de carreras han sido de gusto de los gamers, que utilizan diferentes consolas.

Para nuestro caso, el plano del juego será una matriz, que deberá tomarse de archivos (.txt, .csv, .xml, json) y generar la representación de las pistas de juego y se deben generar las estructuras correspondientes para el desarrollo de la funcionalidad por medio de una interfaz web.

¿Qué se busca con este proyecto?

El objetivo general de este proyecto es desarrollar una mejor herramienta Web para el desarrollo de un juego en línea desde la perspectiva de la programación orientada a objetos, la investigación de estrategias de manejo de concurrencia y multiusuarios; de manera que las partes de la lógica y datos (Backend) que lo enmarcan sean diseccionadas minuciosamente para desarrollar una solución web (Frontend). El Backend deberá desarrollarse con .Net Core (C#) o Node js (Express) aplicando orientación a objetos y el Frontend deberá desarrollarse con React, Vue.js, Angular o Net –AspNet– (manteniendo la separación de proyectos). Puede utilizar un motor de base de datos libre para mantener la información persistente. La información de configuración debe estar en un archivo específico para esta finalidad.

El servidor “gaming” (Backend) y el Frontend se publicarán por medio de las herramientas Cloudflare Tunnel, Ngrok, Localtunnel, Tailscale o similar para que puedan ser accesadas por medio de enlaces públicos (ip pública).

Con esto se busca:

1. Practicar las habilidades de modelado de aplicaciones de software.
2. Ejercitar la toma de decisiones sobre el dominio del problema y de la solución.
3. Aplicar los conceptos del Paradigma Orientado a Objetos en un proyecto programado.
4. Utilizar frameworks y herramientas de tendencia en el mercado.
5. Fomentar el trabajo en equipo.

Proyecto a desarrollar

Se requiere una investigación acerca de las funcionalidades detalladas de los juegos de carreras. Antes de iniciar el trabajo debe analizar por completo el funcionamiento del juego, ya que lo que se desea es reproducir el comportamiento del mismo (juego en línea multiusuario en tiempo real).

El juego debe disponer de las siguientes funcionalidades:

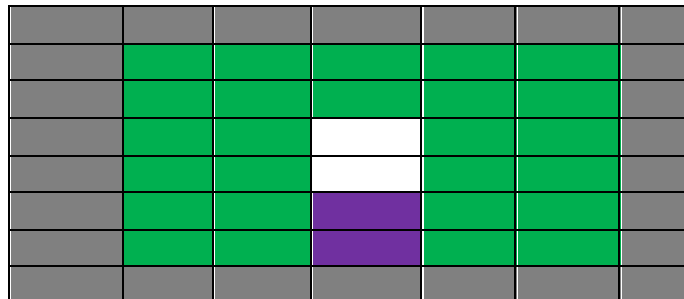
Cliente

Se debe desarrollar la funcionalidad en la aplicación del cliente en la cual el usuario pueda elegir entre tres opciones:

- a) Crear partida
- b) Unirse a una partida (jugar)
- c) Ver ranking

Pista predefinida

El sistema debe tener al menos una pista de juego, con un nombre de la pista y una temática (esto aplica a la hora de “pintar” la pista en la aplicación), la dimensión estándar de 25x25, con una posición de partida y espacios de pista, con al menos espacio para dos carriles. El punto de llegada debe ser el punto de partida por lo que debe existir un camino entre la salida y la llegada.



Autenticación

Para ingresar a las funcionalidades del juego, el usuario deberá autenticarse escribiendo un nickname (no hay gestión de usuarios).

Crear partida

Debe desarrollar la funcionalidad para generar un nuevo juego, donde el usuario podrá personalizar el juego por medio de las siguientes opciones:

- a) Seleccionar el tipo de juego:
 - i. Vs: los usuarios “corren” durante el tiempo que deseen hasta que los todos terminan.
- b) Seleccionar pista:
 - i. El usuario selecciona la pista que desea jugar y la cantidad de vueltas que deben completar y la cantidad de jugadores (mínimo dos).

- c) Una vez creado se generará un código identificador de partida. La partida estará creada por 3 minutos (esto deberá ser un parámetro de configuración), si en ese tiempo no es iniciada (debe “arrancar” el juego), la partida se cierra.

Unirse a juego (partida)

Los usuarios tendrán la funcionalidad unirse a una partida:

se le muestra los usuarios la lista de partidas por iniciar con su información: identificador de juego, la pista, el recorrido según la temática, y los usuarios unidos (deberá irse actualizando conforme se unan). La página actualizará constantemente las partidas y sus usuarios unidos. Debe cargarse la pista desde un medio persistente (seleccionado por el usuario o manejado internamente) y desplegarlo en interfaz según lo indicado en su contenido. Los usuarios estarán a la espera de completar los usuarios de la partida. Los usuarios pueden elegir un vehículo (*).

Una vez que se completen los usuarios en una partida se habilitará un botón para “entrar” al juego, posteriormente al darle entrar, a todos los usuarios se les carga la partida de juego (pista con contrincantes). Para iniciar se debe presionar la tecla u, antes de esto los vehículos no se moverán. Y se darán 3 segundos para arrancar.

Juego:

El área de juego debe disponer de lo siguiente:

- a) Un área para poder jugar o correr, donde inicialmente se muestran los vehículos (al inicio de la pista) y el usuario pueda ir moviéndolos a un espacio válido.

Cada jugador para moverse debe ir cambiando la dirección del vehículo con las letras de dirección (flechas) según corresponde. El vehículo se mueve en la dirección indicada por la flecha un espacio a la vez, siempre que sea un espacio válido. Se puede manejar la opción de desplazamientos mientras se mantenga presionado el botón de flecha en una dirección manejando un límite de movimientos por segundo (configurado).

Cada desplazamiento es un “brinco” de celda. En todo momento se observa la posición propia y la de los “rivales” en tiempo real. La posición de los rivales se actualiza cada 0.5 segundos, o bien, cada vez que varíe.

Si un vehículo tiene una dirección hacia un espacio que no es parte del recorrido de la pista, el vehículo no avanza.

La dirección de **la pista será en sentido a las manecillas del reloj**. *Si un vehículo está avanzando en sentido contrario debe indicarlo.*

- b) Nombre de cada jugador con el vehículo que le corresponde.
- c) Indicar gane: el sistema debe validar que todos los vehículos hayan completado las vueltas o terminado el tiempo e indicar el ganador.

Estadísticas por juego:

Por cada juego se debe indicar lo siguiente:

- a) Nombre de jugadores y su tiempo o posición, ordenado ascendentemente
- b) Pista
- c) Identificador de partida

Ranking:

Se debe mostrar información de los ganadores de cada partida. Por cada juego se debe mostrar la siguiente información:

- a) Nombre del ganador
- b) Tiempo
- c) Pista
- d) Vueltas
- e) Identificador de partida

Puntos Extra

Las siguientes características no son obligatorias, pero se asignarán puntos extras en caso de que se desarrollen.

- Implementar una versión vs tiempo, donde al crear la partida se indica la cantidad de tiempo de la partida y la partida termina cuando se acabe el tiempo o cuando todos hayan terminado. También debe ajustar las estadísticas y el ranking para indicar el tipo de juego. 10 pts
- Se darán **2.5 puntos adicionales** al entregar a más tardar el martes 10 de junio a las 11:55:55 PM el Documento de Requerimientos, ver plantilla suministrada en el Tec Digital. Debe subirse en la documentación llamada "Proyecto Programado 4 (archivos adicionales)" debajo de la carpeta de "Proyectos".
- Habilitar diferentes temáticas de juego (4 pts por temática adicional, máximo 2, siempre que se permita jugar).
- El profesor valorará el desarrollo de las funcionalidades del proyecto, uso de buenas prácticas, usabilidad, diseño, entre otros, y podrá brindar puntos adicionales.

Aspectos técnicos

Se debe generar al menos dos aplicaciones:

- a) Servidor de juego: es quien centraliza las funcionalidades y permite la sincronización de clientes. Debe exponer web services a los clientes para que se pueda generar la interacción y simultaneidad en los usuarios de la partida. Es quien tiene los parámetros de configuración variables.
- b) Cliente: aplicación web con interfaz, que permite exponer el juego a los usuarios, se conecta con el servidor para la generación de partidas y juego en línea. Puede ser accedida por múltiples usuarios. Debe generar mecanismo, en conjunto con el servidor, para dar experiencia de simultaneidad.
- c) Los estudiantes deberán analizar las diferentes posibilidades de herramientas, valorando sus ventajas y eligiendo las que mejor se adaptan.

Deberán utilizar el sistema de control de versiones GitHub, el repositorio deberá ser público o incluir al profesor en el control de acceso de este.

Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, restricciones y **objetivo**.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución y uso**.
3. Pruebas de funcionalidad: incluir *screenshots*.
4. Descripción del problema.
5. Diseño del programa: diagrama de paquetes, de distribución y de comunicación.
6. Librerías usadas: configuración, comunicación, etc.
7. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. Bitácora (autogenerada en git, commit por usuario incluyendo comentario)

Forma de trabajo

El trabajo se debe realizar en grupos de dos.

Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación.

La tarea tiene un valor de **15%** de la nota final, en el rubro de Proyectos.

Desglose de la evaluación de la tarea programada:

1. Documentación interna 2 puntos.
2. Documentación externa 8 puntos.
3. Funcionalidad 80 puntos.
4. Revisión del proyecto (según completitud del proyecto y gestión del tiempo) 5 puntos.
5. Hora de Entrega 5 puntos.

Aspectos administrativos

Debe crear un archivo **.zip** ("PP4_Integrante1_Integrante2.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **programa**, en la primera deberá incluir el documento de *word* o pdf solicitado y en la segunda los archivos y carpetas necesarias para la implementación de este proyecto programado, y/o link en git del repositorio. El archivo **info.txt** debe contener la siguiente información (cualidades):

- a. Nombre del curso
- b. Número de semestre y año lectivo
- c. Nombre de los Estudiantes
- d. Número de carnet de los estudiantes
- e. Número del proyecto programado
- f. Fecha de entrega
- g. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Entrega

Deberá subir el archivo antes mencionado al TEC Digital en el curso de LENGUAJES DE PROGRAMACIÓN GR 60, en la asignación llamada "P4" debajo del rubro de "Proyectos". En la evaluación del Proyecto el rubro de "Hora de Entrega" valdrá por 5 puntos de la nota total del proyecto, según la siguiente escala:

- a. Si se entrega antes de las 11:55:55 **PM** del miércoles 25 de junio de 2025, 5 puntos.

NO SE ACEPTARÁN trabajos que contengan "commits" posterior a esta fecha.

Todo el contenido de cada proyecto debe ser 100% original y en caso de plagio todos los integrantes del grupo tendrán nota cero.

Todos los miembros del grupo deberán participar de la revisión, ya que de lo contrario no se les asignará el puntaje correspondiente.

Referencias

Ngrok. <https://ngrok.com/product>

Top 4 Best Ngrok Alternatives in 2020. <https://www.softwaretestinghelp.com/ngrok-alternatives/>

‘Ngrok’: una herramienta con la que hacer público tu localhost de forma fácil y rápida. <https://sdos.es/blog/ngrok-una-herramienta-con-la-que-hacer-publico-tu-localhost-de-forma-facil-y-rapida>