

Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores
Taller de programación

Especificación del proyecto 2

Profesor Luis Barboza Artavia

I 2021

Tabla de contenido

Introducción.....	3
Especificaciones técnicas	3
Mecánicas de juego	3
Proyectiles.....	3
Niveles.....	4
Puntaje	4
Sonido	4
Pantallas.....	4
Pantalla de inicio	4
Pantalla complementaria o <i>about</i>	4
Pantalla de mejores puntajes	5
Pantalla de juego.....	5
Documentación externa	5
Aspectos administrativos	6
Atributos de evaluación	6
Ensayo sobre ética, equidad y trabajo en equipo.....	6
Evaluación	6

Introducción

Como parte del desarrollo de un nuevo producto en una organización para la cual usted trabaja, se le encomendó a usted la creación de una aplicación prueba de concepto que permita comprobar las capacidades del producto. Mientras usted desarrolla dicha aplicación, la organización decide que sería conveniente realizar otra aplicación similar. Las pruebas a realizar tienen el propósito de evaluar el desempeño en escenarios en que la cantidad de elementos en pantalla es mayor, y no saben si esto puede tener un efecto significativo sobre el desempeño en comparación a otros computadores similares.

Para el momento en que los requisitos para esta nueva aplicación de prueba son definidos, usted ya ha terminado la primera aplicación. Se considera que su experiencia le permitiría completar la nueva aplicación más rápidamente que otro de los trabajadores disponibles, por lo cual se le asigna a usted esta tarea. Se le provee la siguiente especificación de requisitos a cumplir.

Especificaciones técnicas

- Resolución de pantalla a escogencia del programador
- Programa desarrollado en python 3 o posterior
- Uso de Pygame o Tkinter como biblioteca de interfaz gráfica
- Controles con las cuatro flechas del teclado para movimiento
- Biblioteca de sonido vlc-python
- **Temática a escogencia del programador.** Debe ser coherente.

Mecánicas de juego

En forma general, el juego consiste en esquivar proyectiles u obstáculos. Si la temática fuera similar al juego anterior, el jugador podría ser representado por una nave, y los proyectiles a esquivar podrían ser meteoritos.

El jugador controla su avatar utilizando las flechas direccionales del teclado. El objetivo del juego es sobrevivir a varios proyectiles que rebotan alrededor de la pantalla. Entre mayor tiempo sobreviva un jugador, mayor puntaje consigue.

El juego ocurre en una pantalla de fondo estático, el cual debe ser acorde a la temática seleccionada.

El jugador posee en total 3 vidas. Cada impacto de proyectil le restará una vida al jugador. Al llegar a 0 vidas el jugador pierde el juego.

Proyectiles

Los proyectiles son objetos que se mueven alrededor de la pantalla de juego. El movimiento de estos objetos es determinado por una componente vertical y una componente horizontal. Cada vez que un proyectil colisiona con el extremo de la pantalla, invierte el signo de estas componentes y cambia la magnitud de cada uno por un número generado

aleatoriamente. Lo anterior significa que los meteoritos se encontrarán rebotando alrededor de la pantalla en trayectorias relativamente impredecibles.

Al decidir qué rango utilizar para la generación de números aleatorios, recuerde que la prioridad es que el juego sea entretenido.

Los proyectiles no colisionan entre sí. Si colisionan con el jugador, el jugador pierde una vida y el proyectil se destruye

Niveles

El juego consiste de tres niveles distintos. La cantidad de proyectiles a esquivar en cada nivel es mayor que en el nivel anterior. Queda a decisión del desarrollador la esta variación de cantidad. Recuerde que el juego debe ser retador, pero no imposible. En sí, se esperan al menos un nivel relativamente fácil, uno intermedio, y un nivel difícil.

Puntaje

La cantidad de puntos por cada segundo varía según cada nivel. En el nivel fácil, es 1 punto por segundo. En el nivel intermedio, son 3 puntos por segundo. Finalmente, en el nivel difícil, se le confieren 5 puntos al jugador por cada segundo transcurrido. Cada nivel dura un minuto.

Si al finalizar el juego (complete o no el nivel) el puntaje de jugador supera alguno de los mejores anteriores, se le comunica por medio de un mensaje cuál posición obtuvo y con cuántos puntos llegó a dicha posición.

Sonido

Cada nivel deberá tener su propio tema musical acorde al tema escogido. La pantalla principal también tiene su propio tema musical.

Cada colisión de un proyectil contra los extremos de la pantalla causa un sonido de impacto. Este sonido debe ser acorde al tipo de objeto seleccionado como proyectil, por ejemplo: si se usa un meteorito se espera el sonido de una roca chocando, si se usa un cubo de metal, se espera un sonido de un metal chocando.

Pantallas

Toda pantalla que no sea la de inicio debe proveer alguna manera de volver a la pantalla de inicio

Pantalla de inicio

En esta pantalla el jugador puede acceder a cualquier otra pantalla del juego por medio de botones. Además de los accesos a otras pantallas, se muestra una caja de texto y una selección de nivel. La caja de texto (Entry) se usará para permitirle al jugador ingresar el nombre con el que se registrará su puntaje. Los selectores de nivel permitirán empezar por el nivel que el jugador escoja, aunque el valor por defecto será el nivel fácil.

Pantalla complementaria o *about*

En esta ventana se mostrará la información complementaria. Se deben incluir los

siguientes datos:

- País de producción
- Universidad y Carrera
- Asignatura, año que cursa y grupo
- Nombre del profesor
- Versión del programa
- Autor
- Autores de módulos modificados/utilizados por usted
- Instrucciones o datos que considere importantes para el uso del programa

Pantalla de mejores puntajes

Esta pantalla contiene una lista de los 7 mejores puntajes obtenidos. Estos puntajes se leen desde un archivo .txt, el cual es procesado como un archivo secuencial.

Cuando un jugador termina una partida, debe cargar el archivo .txt, comparar los resultados guardados y actualizar el archivo en caso de ser necesario. Si luego de terminar una partida se obtiene uno de los 7 mejores puntajes, esto tiene que mostrarse al volver a la pantalla principal y abrir la ventana de mejores puntajes.

Los puntajes se deben ordenar de forma descendiente utilizando una implementación propia del algoritmo **quicksort**.

No se permite el uso de bibliotecas de manejo de bases de datos o automatización de lectura de archivos secuenciales. Los datos de puntajes deberán ser manejados únicamente utilizando las funciones de acceso a archivos vistas en clase. Lo mismo aplica para bibliotecas de serialización de datos a archivos. Se espera que usted implemente un formato de texto propio.

Pantalla de juego

La pantalla de juego tendrá un fondo agradable a la vista y acorde al tema escogido. Además de los elementos de juego ya abarcados como proyectiles y jugador, deberán mostrarse los siguientes elementos de interfaz gráfica:

- Puntaje
- Tiempo transcurrido
- Nombre del jugador
- Vidas del jugador
- Botón de regreso a pantalla principal (automáticamente aborta el juego, procesar el puntaje es opcional en este caso)

Documentación externa

La documentación externa se hará a través del wiki de *GitHub* y deberá contener las siguientes secciones:

1. Introducción

2. Secciones de código importantes: aquí debe hacer un pequeño resumen de las secciones de código básicas encargadas del movimiento, tanto del vehículo del usuario, como de los proyectiles. También se debe explicar cómo se implementó el algoritmo de quicksort.
3. Resultados: poner y explicar screenshots de las partes más importantes del proyecto.
4. Conclusiones y recomendaciones.

Aspectos administrativos

1. El proyecto será desarrollado en grupos de dos personas. Los estudiantes serán los encargados de crear los grupos; solamente se deberán anotar en [este documento](#).
2. Cualquier intento o consumación de fraude será castigado de acuerdo con reglamento que para tal efecto ha dispuesto el Tecnológico de Costa Rica.
3. **La fecha de revisión será el miércoles 23 de junio a partir de las 9:00am.**
4. Se programará una defensa de su proyecto luego de la entrega del mismo, en el cual demostrará personalmente el funcionamiento del proyecto.
5. En la solución puede utilizar iteración o recursividad indistintamente.
6. Se recomienda el uso de *Tkinter* o *Pygame* para desarrollar la interfaz gráfica.
7. **Es obligatorio el uso de *GitHub* y se debe mostrar el aporte de cada uno de los miembros del grupo en la historia del repositorio.**
8. En la parte funcional debe subir al TEC Digital únicamente un documento en formato .txt con el link al *GitHub* de su proyecto.

Atributos de evaluación

Ensayo sobre ética, equidad y trabajo en equipo

Desarrolle un ensayo, **con una extensión de una página** (no más y no menos), donde desarrolle los siguientes puntos:

- Ética: explique el impacto que tiene en su formación como profesional, la comisión de plagio en las evaluaciones.
- Equidad: explique la manera a través de la cual usted y su compañero se dividieron las tareas de forma que la distribución no perjudicara ni beneficiara a ninguno de los miembros del equipo.

Por otra parte, en un documento elabore un cronograma de trabajo donde describa cada una de las tareas a realizar, el responsable (rol) de cada una (cuál de los dos estudiantes se va a encargar de realizar esa tarea) y el tiempo que esperan invertir en la realización de cada tarea. Además, debe establecer lineamientos adecuados para el trabajo en equipo (metas, roles, reglas, cronogramas, bitácoras, otros).

Evaluación

- Funcionalidad de la aplicación: 80%
- Documentación externa: 20%
- Ensayo sobre ética, equidad y trabajo en equipo: 5% extra
- **Total: 105%**