Space Invaders

Comprensión del Proceso de Desarrollo en Sistemas y Principales Conceptos de Programación a Objetos

Armando Fallas Garro 2019226675

Kevin Calderón Esquivel 20191517479

Instituto Tecnológico de Costa Rica

Contacto: armandofg3937@gmail.com o [kevinec2000@gmail.com](mailto:kevinec2000@gmail.com)

Índice

[Introducción 3](#_Toc6491893)

[Descripción del Problema 4](#_Toc6491894)

[Diagrama UML 5](#_Toc6491895)

[Análisis de resultados 6](#_Toc6491896)

[Bitácora 7](#_Toc6491897)

[Estadística 8](#_Toc6491898)

# Introducción

Para la comprensión del desarrollo en sistemas a un mayor nivel, se debe empezar conociendo los conceptos básicos de programación orientada a objetos.

El proyecto trata de lograr crear una réplica o algo parecido al juego creado por Tomoshiro Nishikado llamado Space Invaders, para esto se debe conocer la función de los archivos JSON y CSV, además de conocer cómo implementarlos en el programa para que puedan hacer su función en el programa sin fallos, pero antes se debe obtener las especificaciones de las librerías Pygame o Tkinter, esto para elegir la que más se acople al proyecto, además de cómo poder descargarlos y utilizarlas para lograr hacer el programa y no solo esto, sino saber los principales conceptos de programación a objetos, ya que; estos logran llevar a cabo muchas cosas necesarias para el juego.

# Descripción del Problema

El juego debe tener mínimo 30 naves invasores, además de una defensora, la nave defensora solo tiene una vida, debe contar con cierto grado de dificultad que irá aumentando dependiendo del nivel (3 niveles en este caso), la nave defensora podrá moverse y disparar, las naves invasoras pueden disparar o tratar de chocar contra la nave defensora y si no lo logran volverán a la formación, llegada a cierta dificultad pueden caer asteroides donde la nave defensora puede destruirlos al igual que hace con las naves invasoras.

Para lograr lo escrito arriba, se debe tener el conocimiento o investigar la programación a objetos, utilizando la biblioteca escogida (Pygame o Tkinter), darle sonido a disparos, muerte de naves, o música ambiental se logra con archivos CSV o JSON, entonces también se debe investigar como poder hacer que funcionen en el programa, y por último pero muy importante comprender el desarrollo de sistemas.

# Diagrama UML

<https://github.com/ArmandoFG/Proyecto-Space>

\*Le das click en la carpeta space, al entrar buscar el archivo llamado Untitled Diagram.drawio

# Análisis de resultados

# Bitácora

|  |  |  |
| --- | --- | --- |
| Actividades | Tiempo Utilizado | Descripción |
| Escogencia de la biblioteca a utilizar. | 2 horas | Se preguntó a tutores, se vieron videos y se debatió cual nos parecía mejor para llevar a cabo el proyecto.  (al final se utilizaron las 2) |
| Investigación UML. | 3 horas | Se buscó el concepto en google, entramos en la página para crear el UML, se pidió ayuda a un tutor y se hizo el UML. |
| Como se utiliza la biblioteca escogida. | 3 horas | El taller realizado por el tutor y un poco de investigación para mejorar y reforzar el conocimiento entendido en el taller. |
| Saber cómo programar objetos. | 5 horas | Investigación en google, videos en YouTube y un poco de ayuda del tutor. |
| Programación del juego (base) | 6 horas | Primero se logró realizar la ventana principal, cerrar la ventana y preguntar si quieres salir, botones. |
| Escogencia de imágenes, canciones en el formato de archivo correspondiente.  Además de integrar algunas al juego | 2 horas y 30 minutos | Se encontró los archivos en el formato correspondiente, además de añadir la imagen de la nave y en la pantalla principal la imagen escogida. |
| Se agrega las nave defensora y las naves invasoras al proyecto (movimiento, disparo, descenso)  enemiga | 5 horas | Con mucha investigación y bastante prueba y error se logra acoplar esta parte del trabajo a la programación. |
| A la nave defensora se le añade el poder disparar y cuando se logra dar a un enemigo que este desaparezca | 4 horas | Se toma un poco como base como se pudo lograr que la enemiga disparara y lograrlo para la defensora. |
| Poder introducir el nombre del jugador y guardarlos en una lista | 3 horas | Como se utilizó otra biblioteca se tenían que hacer ciertos ajustes como botones, una ventana… |
| Poder iniciar el nivel 1, poder hacer que la nave defensora se mueva. | 3 horas | Cierra la ventana de inicio, se logra hacer a pantalla completa, se cargan naves invasoras y aliada, se logra mover, disparar(ambas),etc. |
| Se creó el archivo CSV y también el cambio a JSON | 5 horas | Se duró bastante en este paso ya que no encontrábamos la forma de hacerlo, hasta que se logró. |

# Estadística

|  |  |
| --- | --- |
| Análisis de requerimientos | 5 Horas |
| Diseño de la aplicación y diagrama de clases | 11 Horas |
| Investigación de funciones | 4 Horas |
| Programación | 8 Horas |
| Documentación interna | 1 Hora |
| Pruebas | 3 Horas |
| Elaboración del documento | Horas |
| Total | Horas |

Conclusión personal

Con base en la investigación realizada, los pasos a seguir y cada minuto de esfuerzo, tomamos mucho de esto no solo para programar el juego, si no aprender esos conceptos básicos que nos ayudarán a programar a un nivel un poco mayor, el conocimiento aprendido gracias a este trabajo nos es importante para practicarlo más seguido y no quedarnos estancados en lo que se conoce; ya que, en la programación hay que pulirse lo más que se pueda para así poder aprenderlo sin que se nos olvide como hacer este tipo de programación.

Es un gran repaso de lo que empezamos hacer hace meses, lo que refresca nuestra memoria y nos mantiene siempre en aprendizaje para manejar mejor la forma de trabajar, la comunicación, el trabajo en equipo entre muchas otras cosas que nos ayudarán en un futuro.