Instituto Tecnológico de Costa Rica

Ingeniería en Computadores

Algoritmos y Estructuras de Datos 1

**Proyecto #1: Invaders**

Profesor: Isaac Ramírez Herrera

Estudiante: Jose Andrés Chavarría Vásquez

Carné: 2017094138

Grupo 1

I Semestre – 2018

1. **Introducción**
2. **Descripción del problema**
3. **Planificación y administración del proyecto**
   1. Features:

* Almacenamiento de mejores puntajes con el nombre de cada usuario.
* Nave controlada por el usuario (movimiento y disparo)
* Control de la nave por medio del teclado o con un dispositivo móvil.
* Niveles infinitos.
* Dos tipos diferentes de enemigos (menor y jefe de fila).
* Máximo de vidas para el jugador (si se acaban pierde el juego).
  1. Historias de usuario:

1. Como jugador, quiero controlar una nave para luchar contra otras naves enemigas.
2. Como jugador, quiero que mi puntuación se guarde en una lista de mejores puntuaciones locales, para aumentar la competitividad.
3. Como jugador, quiero sumar puntos al destruir naves enemigas y progresar en la partida.
4. Como jugador, quiero que los enemigos tengan una serie de distintos movimientos o capacidades especiales para evitar la monotonía.
5. Como jugador, quiero que, al ir progresando en la partida, esta se vuelva más desafiante.
6. Como jugador, quiero que la nave tenga un máximo de vidas, para perder el juego en caso de que falle en destruir las naves a tiempo.
7. Como jugador, quiero poder controlar la nave con un dispositivo móvil, como un celular, para tener otras opciones de control de esta.
   1. Distribución por secuencia de uso e importancia:

Secuencia de uso

Destrucción de enemigos y jugador Importancia: Media

Características especiales de las hileras Importancia: Media

Control desde dispositivo móvil Importancia: Baja

Movimiento y disparos del jugador Importancia: Alta

Generar hileras de enemigos Importancia: Alta

Generar naves (enemigos y jugador) Importancia: alta

**Ejecución**

**Final**

**Inicialización**

Importancia

Tabla de puntuaciones Importancia: Baja

**Minimal System Span**

* 1. Minimal System Span:

Como se muestra en el diagrama anterior, el “minimal system span”, o sea, las características mínimas para tener un juego funcional, incluyen los features:

* Generar el jugador.
* Generar enemigos.
* Generar las hileras de enemigos.
* Movimiento y disparos del jugador.

Con las funciones anteriores, se tiene un juego funcional que el jugador puede utilizar, aunque no se hayan implementado el resto de funciones. Es como la versión alpha del desarrollo del juego, es la primera versión.

* 1. Plan de Iteraciones:

Secuencia de uso

Movimiento y disparos del jugador Importancia: Alta

Generar hileras de enemigos Importancia: Alta

**Ejecución**

**Final**

**Inicialización**

Importancia

Generar naves (enemigos y jugador) Importancia: alta

**1. Minimal System Span**

Destrucción de enemigos y jugador Importancia: Media

Características especiales de las hileras Importancia: Media

**2. Iteración**

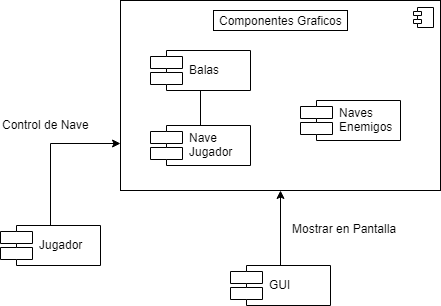
Control desde dispositivo móvil Importancia: Baja

Tabla de puntuaciones Importancia: Baja

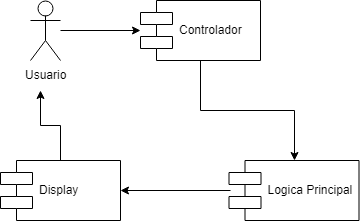
**3. Iteración**

* 1. Descomposición de Historias de Usuario en tareas:

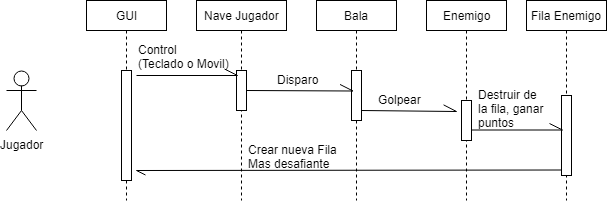
1. Ventana de puntuaciones máximas. Almacenamiento de datos.
2. Manejo, destrucción y disparo de la nave jugador.
3. Movimiento y destrucción de naves enemigas.
4. Aumento de la dificultad.
5. Vida máxima del jugador y pantalla de Game Over.
6. Control de la nave desde un celular.
7. **Diseño:**
   1. Diagrama de Componentes:



* 1. Diagrama de Arquitectura:



* 1. Diagrama de Secuencia:



Se seleccionaron las historias de usuario a, b, e y g, para realizar este simple diagrama de secuencia.

* 1. Diagrama de Clases inicial: