Instituto Tecnológico de Costa Rica

Rubén Salas Ramírez

2017164846

Ingeniería en Computadores

Proyecto #1 - Invaders

CE1103 - Algoritmos y Estructuras de Datos I

Prof. Diego Noguera

Cartago, Costa Rica

Abril 13, 2018

1. Introducción………………………………………………………………………...…2
2. Problema………………………………………………………………………………3
3. Planificación y Administración………………………………………………………..3
   1. Features e Historias de Usuario…………………………………………………...3
   2. Distribución de Historias de Usuario……………………………………………...4
   3. Minimal System Span……………...……………………………………………...6
   4. Plan de Iteraciones………………………………………………….......................6
   5. Descomposición en Tareas……………………………………………………......8
4. Diseño………………………………………………..……………………..................9
   1. Diagrama de Componentes…………………………………………………..........9
   2. Diagrama de Arquitectura…………………………………………………............#
   3. Diagrama de Secuencia…………………………………………………................#
   4. Diagrama de Clases Inicial…………………………………………………..........#
   5. Diagrama de Clases Final…………………………………………………............#
5. Implementación…………………………..…………………………………………....#
   1. Estructuras de Datos Desarrolladas……………………………………………….#
   2. Algoritmos Desarrollados…………………………………………………............#
   3. Problemas Encontrados…………………………………………………................#
6. Conclusión…………………………..………………………………………………...#
7. Bibliografía…………………………..………………………………………………..#

I. Introducción

Space Invaders es un juego de arcade matamarcianos de los años 70´s de los más importantes de la historia. Con una serie de elementos muy simples, a cómo han evolucionado los videojuegos hoy en día, este se mantiene como uno de los favoritos de muchos. Los jóvenes de ahora tienen a disposición juegos en 3D y hasta realidad virtual que demuestran la simplicidad de estos clásicos. Teniendo las facilidades de programación de hoy en día se nos ha puesto un reto de hacer una réplica con ciertas variaciones con una diferente dificultad.

Por medio de los conceptos aprendidos en clase, deberán ser implementados para cumplir con el diseño y solución del juego. Las listas enlazadas y sus variaciones serán una parte fundamental de este, ya que por medio de ellas las hileras de enemigos serán mostradas e implementadas a través de todo este. Por medio del lenguaje de programación con un paradigma orientado a objetos, Java, se llevará a cabo la escritura de código de este proyecto.

Por medio de patrones de diseño se llevarán a cabo diferentes prácticas y soluciones previamente verificadas para un mejor manejo del programa. Gracias a esto se promueve un espacio para analizar cómo se puede implementar esto en un futuro. Aprender buenas prácticas y como desempeñarnos bajo requerimientos preestablecidos, como si fuesen clientes, incrementan la motivación y las ganas de salir adelante.

II. Problema

En este proyecto se procurará diseñar e implementar un juego al estilo de Space Invaders. Este utilizará listas enlazadas en su lógica. El jugador podrá controlar una nave y moverla horizontalmente con el fin de eliminar todos los invasores. Estos serían hileras de enemigos con comportamientos diferentes que intentarán llegar a la parte inferior de la pantalla. Se aplicarán patrones de diseño en la solución del problema.

III. Planificación y Administración

III. a. Features e Historias de Usuario

* El usuario se registra para iniciar el juego.
* El usuario puede utilizar el teclado de la computadora para mover la nave y disparar.
* El usuario puede utilizar el acelerómetro del celular para mover la nave y disparar en la aplicación.
* El usuario mueve la nave horizontalmente en la parte inferior de la pantalla para eliminar a los enemigos invasores.
* El usuario dispara a las hileras de enemigos creadas aleatoriamente por el juego para que los enemigos se muevan hacia el centro yendo hacia la parte de debajo de la pantalla.
* El usuario lee estadísticas importantes en la pantalla y en el celular para mantenerse informado de la partida.
* El usuario gana vidas cada cierta cantidad de puntos para continuar jugando por más intentos.
* El usuario sube de nivel en el juego para que las hileras de enemigos se hagan más rápidas.
* El usuario recibe las estadísticas finales del juego y los tres puntajes más altos al perder el total de sus vidas.
* El usuario tiene la opción de reiniciar el juego, al perder, para jugar de nuevo.

III. b. Distribución de Historias de Usuario

|  |  |  |
| --- | --- | --- |
| Feature | Criticalidad (Valor) | Frecuencias de Uso |
| El usuario se registra para iniciar el juego. | Bajo | Medio |
| El usuario puede utilizar el teclado de la computadora para mover la nave y disparar. | Alto | Alto |
| El usuario puede utilizar el acelerómetro del celular para mover la nave y disparar en la aplicación. | Alto | Medio |
| El usuario mueve la nave horizontalmente en la parte inferior de la pantalla para eliminar a los enemigos invasores. | Alto | Alto |
| El usuario dispara a las hileras de enemigos creadas aleatoriamente por el juego para que los enemigos se muevan hacia el centro yendo hacia la parte de debajo de la pantalla. | Alto | Alto |
| El usuario lee estadísticas importantes en la pantalla y en el celular para mantenerse informado de la partida. | Alto | Bajo |
| El usuario gana vidas cada cierta cantidad de puntos para continuar jugando por más intentos. | Medio | Medio |
| El usuario sube de nivel en el juego para que las hileras de enemigos se hagan más rápidas. | Medio | Alto |
| El usuario recibe las estadísticas finales del juego y los tres puntajes más altos al perder el total de sus vidas. | Bajo | Bajo |

III. c. Minimal System Span

Repartidas las tareas, se determina la primera iteración de funcionalidades. Se podrá tener los elementos críticos del programa para que este posea las funcionalidades básicas que el cliente esperaría. Se determinaron estas funciones críticas:

* El usuario mueve la nave horizontalmente en la parte inferior de la pantalla para eliminar a los enemigos invasores.
* El usuario puede utilizar el teclado de la computadora para mover la nave y disparar.
* El usuario dispara a las hileras de enemigos creados aleatoriamente para que enemigos se muevan dentro y no lleguen a la parte de debajo de la pantalla.
* El usuario lee estadísticas importantes en la pantalla y en el celular para mantenerse informado de la partida.

Estas serían las primeras funciones que se pondría a trabajar para el programa para que su primera iteración prioritaria sea completada. Una vez lista, se continuará con la siguiente iteración descrita en el plan de iteraciones.

III. d. Plan de Iteraciones

* Primera Iteración
  + Duración Aproximada: 2 semanas y media
  + Se crearán los objetos principales como los enemigos, los jefes, sus respectivas hileras y la nave principal para poder correr el juego con sus funciones básicas. La nave se podrá mover horizontalmente y disparos saldrán de esta para destruir los enemigos.
  + Se crearan aleatoriamente diferentes hileras de enemigos y jefes que intentaran llegar hasta la parte de debajo de la pantalla. Mientras el juego corre el usuario podrá leer estadísticas que serán actualizadas periódicamente para su información.
* Segunda Iteración
  + Duración Aproximada: 1 semana y media
  + Durante esta segunda iteración, se agregan otros detalles a la implementación del juego. Se definen niveles donde el usuario puede aumentarlos al ganar cierta cantidad de puntos. También, se creará la posibilidad de que la nave tenga una cierta cantidad de vidas que el usuario puede ir ganando también por su puntaje
  + Se creará una sencilla aplicación en el celular que podrá controlar a la nave por medio del acelerómetro. En ella habrá un botón en el cuál se podrá disparar. Las estadísticas e información que sale en la pantalla, saldrá al mismo tiempo en esta.
* Tercera Iteración
  + Duración Aproximada: 1 semana
  + Por último se hará un registro donde los jugadores ingresan su nombre y todas sus estadísticas son guardadas en un archivo o una base de datos. Estas serán guardadas y actualizadas cada vez que termine el juego. También, cuando la nave sea destruida y ya no tenga más vidas disponibles, se mostrarán las estadísticas finales de la partida en una última ventana. Podrá terminar el juego o volver a iniciarlo

III. e. Descomposición en Tareas

* El usuario se registra para iniciar el juego.
  + Se tomará el String con el nombre que sea ingresado en un TextBox y se crea un objeto Player con su respectivo Ship. Se crean las primeras hileras de enemigos, y el juego comienza.
* El usuario puede utilizar el teclado de la computadora para mover la nave y disparar.
  + Por medio de un KeyEvent, las teclas de izquierda y derecha harán que la nave se mueva para su respectivo lado en el programa. La tecla de espacio hará que se cree un disparo hacia arriba y se verificará si el enemigo es destruido o no.
* El usuario puede utilizar el acelerómetro del celular para mover la nave y disparar en la aplicación.
  + Con Android Studio se creará y se diseñará la aplicación que luego se conectará con el juego por medio de un servidor para servir como control gracias a un botón y al acelerómetro del celular.
* El usuario mueve la nave horizontalmente en la parte inferior de la pantalla para eliminar a los enemigos invasores.
  + Cuando se dispare, un método creado verificará si el disparo llega a tocar al enemigo, lo elimina, y según corresponda por la hilera, se reacomodará o se eliminará.
* El usuario dispara a las hileras de enemigos creadas aleatoriamente por el juego para que los enemigos se muevan hacia el centro yendo hacia la parte de debajo de la pantalla.
  + A las hileras se les creará un movimiento constante hacia abajo y hacia los lados hasta que lleguen a una posición igual a la nave, que hace que el juego termine.
* El usuario lee estadísticas importantes en la pantalla y en el celular para mantenerse informado de la partida.
  + Cada vez que un enemigo sea destruido o golpeado, el puntaje, el nivel, la cantidad de vidas, y cuales hileras están actualmente y por salir serán actualizados.
* El usuario gana vidas cada cierta cantidad de puntos para continuar jugando por más intentos.
  + Cada vez que el puntaje llegue a un múltiplo de 1000 se le otorgará una vida más, tendrá un máximo de tres en total.
* El usuario sube de nivel en el juego para que las hileras de enemigos se hagan más rápidas.
  + Conforme se sube de nivel, el intervalo de tiempo de la animación de las hileras llegando hasta abajo, se hará más pequeño. Provocando esto que las hileras se hagan “más rápidas”.
* El usuario recibe las estadísticas finales del juego y los tres puntajes más altos al perder el total de sus vidas.
  + Cuando se destruya la nave se verificará que tenga vidas disponibles. Si no las tiene, el juego terminará, se cambiará la ventana y de las estadísticas obtenidas a lo largo del juego, serán impresas en esta. Del archivo o base de datos que guarda los puntajes de otros jugadores, se darán los tres mejores y se actualizará si el actual es mayor a ellos.
* El usuario tiene la opción de reiniciar el juego, al perder, para jugar de nuevo.
  + En la misma pantalla se creará la opción, por medio de un botón, de reiniciar el juego, o bien, se terminará este.

IV. Diseño

IV. a. Diagrama de Componentes

-

IV. b. Diagrama de Arquitectura

-

IV. c. Diagrama de Secuencia

-

IV. d. Diagrama de Clases Inicial

-

IV. e. Diagrama de Clases Final

-

V. Implementación

-

V. a. Estructuras de Datos Desarrolladas

-

V. b. Algoritmos Desarrollados

-

V. c. Problemas Encontrados

-

VII. Bibliografía

* Notodocodigo. (2018, 16 junio). Diagrama de componentes y diagrama de distribución. Recuperado 9 de abril de 2018 de http://www.notodocodigo.com/blog/diagrama-de-componentes-y-diagrama-de-distribucion/
* Patton, J. (2005, enero). How You Slice It. Recuperado 03 marzo, 2018, de https://jpattonassociates.com/wp-content/uploads/2015/01/how\_you\_slice\_it.pdf
* Oracle. (2010). Arquitectura en capas aplicada a los componentes del sistema. Recuperado el 9 abril de 2018 proveniente de https://docs.oracle.com/cd/E19528-01/820-0888/aaubd/index.html