Dosier RenegadeKlingon

with2balls

Autor: Angel Baltar Diaz

Índice general

Índice de figuras

Capítulo 1

El juego

En este capítulo presentaremos el juego, su temática, estilo y cual es la concepción general del título, que es lo que hace hasta el momento y cuales son en general las líneas de trabajo que están planteadas.

1.1. Temática del juego

Este juego se basa en la típica temática de guerra entre naves espaciales, es un juego estilo R-type en el que el jugador maneja una única nave y debe destruir ejércitos enteros de naves enemigas, dentro de este estilo de juego la temática en si está abierta, no hay una historia concreta en la que el juego se base por el momento.

Se ha planteado desde un principio una temática basada en el universo star Trek en el que el protagonista es un klingon que toma una nave por su cuenta y se lanza a conquistar territorios enemigos, sin embargo como ya decía la temática esta completamente abierta mientras se mantenga el estilo de juego. Si en algun momento se quiere prescindir de la temática basada en star trek por temas de licencias o por cualquier otro motivo, no habría problema en hacerlo, además tecnicamente es simplemente cambiar imágenes de naves, fondos de los escenarios,

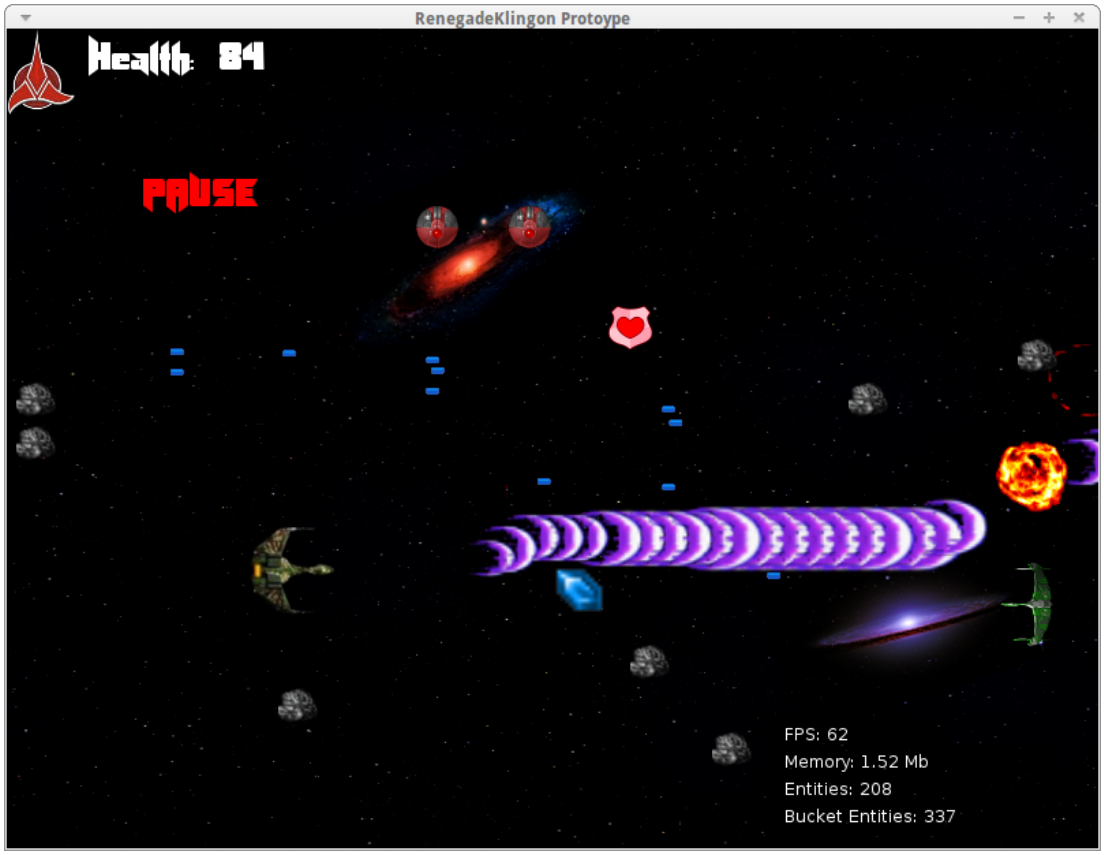

Figura 1.1: Captura de pantalla del juego

logos etc.

1.2. Objetivos

Este juego surje como un simple hobby, he estado dedicando tiempo libre a su construcción, en general las partes de programación no me plantean excesivos problemas, sin embargo tengo lagunas muy importantes en el apartado gráfico porque no sé lo suficiente de diseño gráfico. Tal como he planteado el juego los objetivos que tengo para el son los siguientes:

- Juego en 2d estilo **R-type**[3].

- Niveles diseñables y extensibles sin necesidad de programar (basados en generación mediante Tiled).
- Armas intercambiables que se puedan ir consiguiendo durante el juego.
- Sistema de juego basado en nivel de salud combinable con número de vidas.
- Soporte como aplicación de escritorio para Mac, Windows y linux
- Soporte para dispositivos móviles
- Soporte para jugar a través de Web

Algunos de estos objetivos están actualmente cumplidos, otros son en principio tecnicamente viables y de otros simplemente aun no he mirado como pueden hacerse, en otros capítulos entraremos más en detalles técnicos sobre todas estas cuestiones.

Capítulo 2

Consideraciones técnicas

En este capítulo hablaremos sobre como se encuentra tecnicamente el juego, repasaremos los objetivos planteados en el capítulo anterior y sobre cada uno evaluaremos su estado.

2.1. Juego 2d estilo R-type

Este objetivo es el principal del juego, es el corazón del mismo, conseguir un estilo de juego 2d con una jugabilidad similar a la de los clásicos R-type.

Para conseguir este objetivo ha tenido que programarse el framework base del videojuego, este se basa en una clase llamada Space que no es mas que un simulador del espacio en el que se desrrolla la acción, es decir controla diversas acciones como son:

- Actualización de todos los objetos del escenario en cada frame.
- Sistema de colisiones entre objetos, decide si por ejemplo una bala debe colisionar con una nave provocando una explosión. El sistema de colisiones esta basado en Buckets para mayor rendimiento.

- Dibujado de todos los objetos en el escenario, incluyendo soporte para **parallax** [3] en varios niveles o planos. Este soporte está implementado actualmente pero es bastante mejorable.
- Control de objetos fuera de límite del escenario
- Control de la aparición de objetos al desplazarse el jugador hacia adelante en el mapa

Todo este framework esta actualmente construido y funcionando, no obstante como es el núcleo fundamental del juego esta sujeto a cambios frecuentes, y dependiendo de la dirección en la que se realice el trabajo futuro tendra que ser actualizado, para la implementación de nuevas funcionalidades, temás de rendimiento etc.

2.2. Niveles diseñables y extensibles

Uno de los objetivos planteados desde un principio es que los niveles o fases del juego no vayan directamente programados, sino que sean archivos que el juego sea capaz de cargar. Esto supone una tremenda ventaja ya que el juego se hace extensible por definición simplemente añadiendo más mapas o niveles, no obstante también tiene desventajas como cierta pérdida de flexibilidad al tener que pensar todos los objetos y demás de forma que puedan ser diseñados y cargados en mapas.

El sistema de mapas está actualmente ya implementado, lo que se refiere al soporte básico, el mayor trabajo que falta en el juego y que afecta a los mapas es basicamente implementar cada vez más y más enemigos, armas, minas, y en general objetos cargables desde mapa, para que los diseñadores de niveles del juego tengan un amplio avanico de posibilidades al diseñar sus mapas.

El diseño de mapas se basa en un programa llamado **Tiled** http://www.mapeditor.org/. Este programa se basa en una serie de imágenes (Sprites) y en

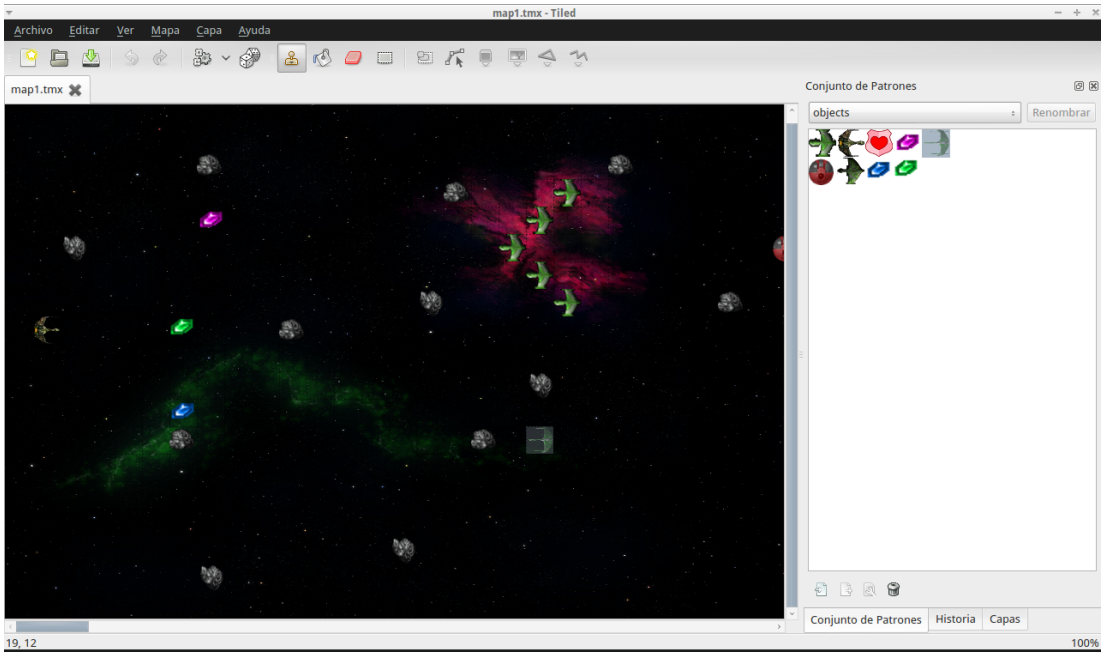

Figura 2.1: Captura de pantalla del programa de generación de mapas

su combinación en celdas para generar los mapas, realmente lo que se genera es un archivo XML que posteriormente el juego cargará cargando con el las imágenes asociadas.

En RenegadeKlingon existe una manera particular de generar los mapas, las imágenes aplicadas en Tiled no son directamente las cargadas por el juego, sino que existen 2 imágenes, una para el diseño de los mapas y otra que es la que el juego cargará finalmente y que va asociada a la primera y representa lo mismo. Por este motivo los diseñadores de mapas, deben probarlos sobre el juego para darlos por finalizados ya que puede que algunas zonas del mapa no queden en el juego tal como se ven en Tiled.

Esta forma de trabajar esta motivada por algunos problemas técnicos que surjen al cargar objetos que ocupan mas de un **tile**[3]

2.3. Sistema de juego basado en nivel de salud y número de vidas

Actualmente está implementado un sistema de juego basado en nivel de salud, que se decrementa con los disparos recibidos o en general colisiones y puede incrementarse al recoger objetos de salud, no está implementado un sistema basado en número de vidas, pero su implementación sería relativamente sencilla y abordable rapidamente.

2.4. Desplegado de la aplicación sistemas soportados y sistemas objetivo

Actualmente el juego se despliega sobre tres sistemas, Mac, windows y linux, para estos tres sistemas el juego es una simple aplicación de escritorio.Se detalla para cada plataforma:

- Mac: Tras el correcto empaquetado se proporciona un .app que tras instalarse da acceso al juego.
- Windows: Tras el correcto empaquetado se proporciona un .exe que es el ejecutable del juego.
- Linux: Tras un simple empaquetado se proporciona un .love que ejecutado por la aplicacion para linux love2d ejecutará el juego.

Existe además un sistema de despliegue automático basado en un script shell que basicamente se encarga de las siguientes tareas:

- Empaquetado del juego para cada plataforma
- Subir cada empaquetado a github en el repositorio llamado RenegadeKlingonDeploy: Para cada plataforma se sube sobre un branch independiente

- Hacer merge entre los branches de cada plataforma y subir dicho merge sobre el branch master para ofrecer la posibilidad de descargar el juego para las tres plataformas a la vez.

Tener el sistema de despliegue automatizado supone una gran ventaja ya que si realizamos cualquier cambio sobre el juego podemos desplegar dicho cambio inmediatamente, y la gente que a partir de ese momento se descargue el juego tendra el último cambio que hemos añadido.

Migrar el juego a Mac y Windows ha sido relativamente sencillo, simplemente son cuestiones de como empaquetar el entregable final que maneja el script comentado anteriormente. Las siguientes plataformas objetivo son dispositivos móviles y desplegado directamente sobre web para ser jugado online, detallamos a continuación ciertas ideas técnicas sobre como llevar a cabo estas migraciones:

- Android: En principio, pensaba estudiar el proyecto love-native-android https://github.com/hagish/love-native-android. Muy probablemente habría que mejorar el juego en temas de resolución de pantalla y puede que aparecieran bugs al jugar sobre android. Además habría que estudiar temas de licencia y leer con detalle la licencia del proyecto love-native-android.

- Iphone: Por el conocimiento que tengo hasta ahora no es técnicamente viable, no hay proyectos que migren automaticamente, y por las discusiones en foros hacer un proyecto a tal fin es una tarea muy compleja.Para llegar a Iphone podrían plantearse vias alternativas como jugar via web

- Web: Existe un proyecto llamado love-webplayer https://github.com/ghoulsblade/love-webplayer/wiki que pretende ser un player de love2d sobre javascript, si ese proyecto funciona seria sencillo poner disponible el proyecto via web, también existe algun proyecto similar pero sobre html5, simplemente habria que estudiar estos proyectos y ver cual funciona mejor con el juego.

Capítulo 3

Glosario de Términos

- **R-type** Saga de juegos arcade de naves espaciales, género matamarcianos, que comienza en 1987 y saca multitud de títulos como R-Type, R-TypeII, R-Type Leo...
- **parallax** Sistema de movimiento de objetos en el que el fondo se mueve a menor velocidad que los objetos en planos más cercanos creando una cierta sensacion de 3D sin realmente existir 3D en si mismo.
- **tile** Unidad básica en el diseño de mapas, es una celda del mapa, el objeto más pequeño que puede cargarse es como mucho de tamaño 1x1 tile.