Nuestro proyecto esta basado en el famoso juego de “maquina” Metal Slug.



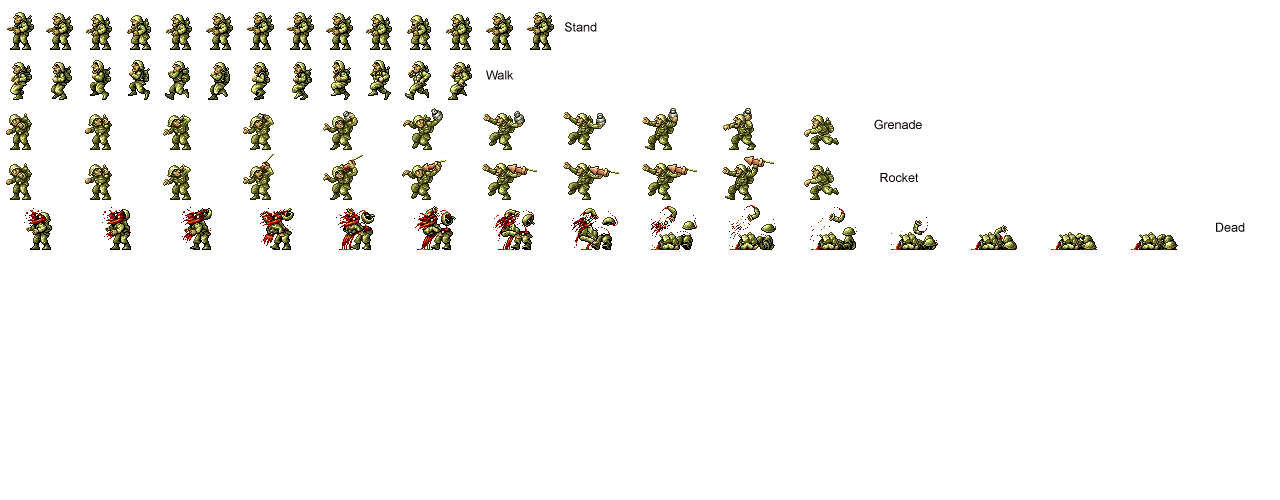
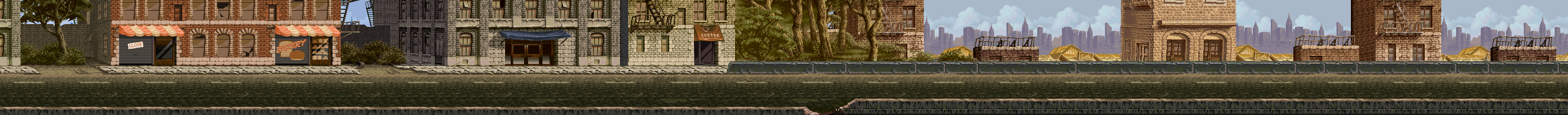
Es un juego en 2D en el que un personaje o dos (seleccionables),

“Marco Rossi” “Tarma Roving”

tienen que matar a los soldados enemigos (1 punto por enemigo derrotado) para luego acabar con el enemigo principal, de ahora en mas “Boss” (100 puntos por Boss derrotado a cada jugador). Nosotros incluimos 3 niveles, con sus diferentes escenas, personajes, sonido y su respectivo Boss.

Cada soldado enemigo tiene 4 estados, caminar hacia un jugador, quedarse quieto, disparar un proyectil con tiro oblicuo o disparar un proyectil en linea recta.

Cada jugador cuenta con una munición inicial de 200 balas de ametralladora, la cual permite disparar manteniendo presionada la tecla de disparo. Al acabarse las balas el jugador deberá presionar una vez por cada bala que quiera disparar (esto se hizo para dar un poco de dificultad al juego, igualmente al pasar de nivel se restaura todo tipo de munición). Cualquiera de estas dos formas solo le quita “1” punto de vida a los soldados, y solo colisiona con el primero, la vida normal de un soldado es “2”, sin embargo cada jugador posee inicialmente 50 granadas, las cuales hacen “2” de daño y además colisionan con todos los enemigos que encuentre al explotar (esto en realidad fue inicialmente un bug, ya solucionado, de la pistola que luego lo implementamos como efecto para la granada), lo cual da ventaja por sobre la ametralladora en grupos de enemigos. El juego se termina al quedarse sin vidas o al acabarse el tiempo.



Entre otras habilidades del jugador, están las de saltar (lo cual nos llevo trabajo hacer que sea en forma “parabólica” al igual que el disparo de granadas del jugador y de los enemigos, entre otros), y de poder recorrer el fondo en caso de estar posicionado mas de media pantalla y no se encuentre ningún enemigo vivo en ella.

El juego se encarga de cargar y guardar los puntos de cada jugador en el archivo “highscores.txt” al finalizar el juego, luego de que cada jugador seleccione su nombre en la pantalla que le aparecerá cuando no queden jugadores vivos. Al jugar en modo multijugador, el segundo jugador deberá mirar la pantalla del jugador numero 1 para poder jugar, sin embargo en su pantalla aparecerán los botones del teclado y cambiaran de color al ser presionadas para indicarle que se están recibiendo datos del teclado correctamente.

El segundo jugador también debe saber la dirección ip del primer jugador (ya que es una conexión TCP) para poder conectarse mediante consola. Al conectarse le saldrá una ventana negra esperando que el jugador 1 (servidor) elija los personajes, luego de la selección de personaje el servidor le permite empezar a enviar datos al segundo jugador (cliente) y recién luego de eso le aparece la ventana con las teclas antes mencionadas.

En general el código esta organizado en estructuras para cada “objeto”, jugador, enemigo, balas del jugador, del enemigo, Boss, muertes enemigas, entre otros. Luego también se esta dividido en varios archivos “.c” con funciones relacionadas a cada objeto, hacia el jugador y su refresco de sprites y dibujo en la ventana al igual que los enemigos, hacia las balas y colisiones de ellas y la de los enemigos, hacia la tabla de puntuaciones, las funciones del servidor para conectarse con el cliente, explosiones y demás.

Entre otros problemas que nos encontramos en el desarrollo del juego (además de los ya mencionados) fueron el hecho de decidir cual era la base solida del juego como para empezar a agregar funcionalidades y poder trabajar en equipo, modificando/agregando diferentes funciones. El empezar a entender como funciona Allegro y familiarizarse con sus funciones y buscar “utilidad” de ellas, como por ejemplo, si el juego no encuentra una imagen que necesita ser cargada, además de informarlo por consola, gracias al native dialog de Allegro hicimos que aparezca una ventana con información del error para informar al usuario sin necesidad de una consola, aprendimos mucho gracias a la Wiki de Allegro y varios ejemplos de códigos que vimos. Empezamos programando cosas sencillas, moviendo objetos en pantalla y como detectar una tecla presionada o solo cuando se presiona y/o suelta, luego seguimos con sprite sheets de paginas de internet y de a poco fuimos agregando lo que aprendíamos a hacer. También fue desafiante el hecho de usar sprites de un mismo personaje con diferentes tamaños, como es el caso del soldado enemigo, al lanzar un proyectil su frame es mas grande y esto presentaba dificultades a la hora de su movimiento (solucionado con una variable de ajuste que se suma/resta cuando se esta ejecutando dicha acción). Otro problema que tuvimos fue el movimiento en conjunto del fondo, las balas y posiciones enemigas mientras el jugador recorría el terreno (ya que esto producía en los soldados que estén parados pero sin embargo el piso se moviera, al igual que con las balas se producía un efecto de menor y/o mayor velocidad), cosa que decimos que era conveniente que no se pueda avanzar mientras halla enemigos vivos, lo cual también agregaría un poco mas de dificultad al juego para que no sea tan fácil y pero sea mas divertido. Cabe destacar que en varias ocasiones nos ayudo mucho el ddd (Data Display Debugger) en solucionar bugs o errores del código, sin embargo en otras no tanto.