

GameBuilder | Project

Requerimientos de software

Cristian Rivera López

El proyecto de GameBuilder tuvo sus primeros inicios entre los meses junio-agosto del año 2013, aun así no había sido contado como un proyecto en ese entonces, nació como una simple idea, un programa que creara mapas de un juego *sidescroll* y este mapa se albergara en un archivo XML ahora el proyecto tiene como propósito algo mas elaborado, consiste en proporcionar herramientas que colaboren a la creación de videojuegos tipo *sidescroll* para usuarios de nivel programación medio en java.

Lo pueden usar usuarios de Linux, Mac y Windows, compactado en una aplicación con extensión “jar”. La aplicación contiene un creador de mapas en el cual se usan tantas capas como se quieren, se colocan objeto que pueden o no interactuar se pueden crear objetos o no, creador de objetos (tiene la opción de crear el objeto en código para que explote la cantidad de objetos que se puedan crear y no se vea limitado por la opciones que ofrece el programa), creador de entes, con la capacidad de crear los algoritmos de ataque, movimiento, defensa tanto visibles como en código, además contiene una especie de kernel que puede leer los

productos creados, emularlos y pueda ser utilizado en muchos otros proyectos. Los anteriores se analizarán por separado seguidamente:

SideScroll MapMaker: Crea mapas de una partida, colocando objetos creado en las siguientes clasificaciones: superficies, transportadores, entes (también permite colocarlos aleatoriamente, eliminarlos, etc), punto inicial y puntos de guardado, interactivos (los cuales tienen que programarse por aparte, pues el programa obviamente no posee tanto en su galería). El mapa puede probarse.

ObjetsMaker: Crea objetos de las siguientes clasificaciones: Objeto de mapa (objetos que interactúan o no en el mapa), objeto de ente (objetos que puede usar un ente enemigo o el personaje mismo); los objetos de ente además se clasifican en armas, interactivos (deben programarse) y curadores.

EntityMaker: Crea entes enemigos o personajes, con esta aplicación pueden manejarse los sprites de ataque, añadirse o quitarse ataques, dar objeto, añadir condiciones

de ataque, guardar los cambios por aparte o sobre escribirlos, entre otras idea secundarias.

Kernel: Actúa como una biblioteca que puede importarse dentro de algún proyecto de juego el kernel tiene la responsabilidad de pintar los objeto en pantalla, cargar los menús que el

programador indica, los objeto, mapas, conversaciones, etc. por otro lado dentro del proyecto GameBuilder tiene la tarea de emular cada uno de los objeto cuando se deba, por ejemplo emulación de ataque con o sin enemigo.