Literature Review

Tile-Based Game Design – Springer Link

The chapter in this book relevant to the project is about tile-based game design. It begin with the advantages of tile-based games including:

1. Array Storage
2. Collision Detection
3. Simplified AI
4. Efficient Use of Graphics

Each of these will be used during the project and will be explained when relevant. The chapter then goes onto a step-by-step way of creating a simple tile-based world starting with the array, and ending with character movement and collision. The following is the basics of tile-based game creation, and references directly from the book will be used.

“Tiles are rectangles that contain the graphics” in the game. They “can be any height or width, but their sizes will place certain constraints on the dimension of the stage.” Tiles are usually the same dimensions throughout the game, “computers handle bitmap sizes that are multiplies of 2 very efficiently” and so 64x64, or 32x32 are popular.

Tile sheets are used in order for one sprite image to be needed as appose to multiply, and this is possible because the one large image contains all the needed sprites in the game. “The images in the tiles are all different shapes and sizes. The dimension is only the *maximum* *size* that a tile should be.” Tile sheets are useful because an entirely different image isn’t needed if the sprite changes, just the coordinates of the sprite sheet.

Due to every single tiles needing the same information, such as location, items on the tile etc., “it makes a lot of sense to create a class to store this information for every tile in the game.” This is called the tile model, because it is the baseline/standard example of a tile.

The next step is to put the map into the game. The map is simply the tiles of the world. “To build the game world, “a “nested *for* loop that simulates the game map’s grid” needs to be created. The computer will go row by row, column by column, creating the map out of the tile model class. Due to the grid of the map, each tile has an easy to identify coordinate which makes manipulating the tiles very straight forward.

Adding game characters and layering maps to the next step. Another map is created and this is the foreground map. “The platforms and the sky are in the background and the “character “is in the foreground.” The background map is rendered first, then before the foreground map is rendered, the maps are compared. If there are conflicts then the foreground tile cannot be rendered. For example, a character can occupy the same tile as sky, so it gets rendered fine, but if it the same tile as a wall of floor, there is an issue. If the foreground sprite has transparency, the background image should be visible behind the foreground image, which is often good.

The chapter then carries on with jumping and platforming specific ideas but these are not relevant for this project, and so will not be commented upon. This beginning style of tile map creation is a very good one and it will be the style used for the initial creation of the world.