The technical document containing the description of how each crucial algorithm works.

IT PAT 2018

Technical Document

Ralph McDougall

Contents

Documentation

The signatures of each method can be found in the code itself. A brief description of what each method and field is for is next to the definition of the method and field. If all of the methods and fields were to be included in this document, the size of this document would be impractical.

Algorithm Description

Map Generation

When a new map gets generated, cut the grid of tiles into a grid of map tile segments. Each map tile segment is a square of tiles. Pick a random location in the map to be the starting location. Branch outwards to neighbours of the starting position. At each point in the algorithm, a random node is chosen so that it does not have more than one neighbouring chosen node. This ensures that the maze that is constructed does not have any loops in it, but it still has some order in it to allow long corridors to get constructed in the maze.

Once the map tile segment grid has been constructed, 3 random dead-ends are chosen to be the end area, item area and NPC area. A new item and NPC are placed at their appropriate places. From this map tile segment grid, the tile grid is determined. Possible formats for each map tile segment is given in a text file. Depending on whether a player may walk in a given map tile segment, an appropriate map tile segment layout is chosen from the possible map tile segments.

This allows there to be random maps every time a new map is generated while ensuring that the player can get to every walkable location in the map.

Enemy Path Finding

From the player's location, branch outwards. For any node in the process, move to its neighbours and use the location relative to the given node to determine what direction to move from the neighbour to get to the player as quickly as possible. The only neighbours that should be visited are those that shorten the path to the player.

It should be recorded which nodes are in line with the player so that it is possible for the enemies to know when they can fire at the player.

Quest Handling

When a quest is created, it is assigned a target, type, target count and reward. The type of quest is for example whether it is a quest to find a certain item or whether it is a quest to kill a certain enemy. The count of the quest is the number of times the target of the quest needs to be found. The target is the name of the item to be found or the name of the enemy to be killed.

Whenever an item is picked up or an enemy is killed, all of the quests get checked to see if their counts need to be increased. If a quest gets increased and the target count is reached, the player is given the reward XP.

When the quests are to be saved to the database, it is first checked if the quest has been completed yet. If it has, it is not added to the database.

Minimap updating

The minimap gets passed the location of the player in the map. The minimap then checks the identity of the map tile segments in the area surrounding the player's coordinates. If a segment is walkable, it sets the colour of that area on the minimap to black and white if it is walkable. The Points-of-Interest were recorded when the minimap was created so that the player knows what they need to work towards.