****

**Tyrant’s Realm**

Map Creator Requirements Document

By Mark Heller

Copyright© 2010

Version 1.0 (draft)

July 31, 2010

1. Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Modifier** | **Date** | **Description** |
| 1.0 | Mark Heller | 7/31/2010 | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Overview

The map builder tool will allow the level designer to create maps for the Tyrant’s Realm fantasy web-based RPG MMO dungeon crawling feature. It will be broken into loading and saving and creating levels and level modules. It will also allow the user to generate any number of base maps from randomization of a specified number of modules. A map is an entire level whereas a module is generally a room or hallway (a small piece) that can be merged with others to create a dungeon.

1. New Map / Module / Quest Creation
   1. Create a new project: It needs to allow the creation of a new project file. Only one file is open at a time.
      1. If an edited and unsaved file is already open, then prompt for saving it or not.
      2. Prompt the user for a file name.
      3. Prompt the user if it is a map, module or quest.
      4. Prompt the user for x and y coordinates on the size of the level (for map and module).
      5. Prompt the user for dungeon type (of the 10 possible) (for map and module) or quest start location (town building)
      6. Prompt the user for the dungeon level (if a map).
      7. Prompt for the quest id # (if quest)
      8. The project filename will be either name.map, name.mod or name.qst as per the answer in 3.1.1.3.
      9. Needs a 2D top-down view that will be grid-based. Grid lines should be shown (for map and mod).
         1. The grid will have an x,y coordinate value shown on the sides with 0,0 in the lower left.
      10. Needs a wire-framed 3D 1st person view (for map and mod).
          1. (optional) Show models and textures if possible. Provide basic ambient and spot lighting as if a torch were lit with first person walkthrough.
2. Map / Module / Quest Opening
   1. Open file menu should be presented allowing the user to browse to the file they wish to open.
      1. If an edited and unsaved file is already open, then prompt for saving it or not.
      2. The file should load into the active 2D/3D windows or quest view.
3. Map / Module / Quest Saving
   1. From the File menu, a user should be able to save and SaveAs with standard options for browsing. However, the file will automatically save with the extension .map for map, .mod for modules or .qst for quest.
4. Map / Module / Quest Editing
   1. Set walls (for map and module only)
      1. Option needs to be given to set the cursor to set walls, doors, arches, windows and secret doors.
      2. There should be an option for style, to allow different types of walls, doors, secrets and archways.
      3. There will be a field for textures.
      4. The user will see preset ones for the dungeon type and general for the game.
      5. The user can turn off collision on any given wall.
      6. Left clicking on a grid line will add the appropriate wall type.
         1. If a wall, door etc already exists on the spot, it will override with what the current settings are.
      7. Provide an option to identify connection point for modules.
   2. Set Model (for map and module only)
      1. Provide a submenu to identify if the user is linking the model to a room or wall.
      2. A list of models will be provided to select based on the dungeon and general options.
      3. The user can turn off collision for any given model.
      4. There will be a 9 point location value to be set. For wall, this is the location on the wall where the item is attached or leaned against. With room, it is where in the room and a second 1-3 value is used to identify how high (1 is on the floor and 3 is hanging from the ceiling). Note that room edge numbers will not quite be against the wall.

Format is:

Location Height

(Room or Floor) (Floor only)

1 – 2 – 3 3

| \ / | |

4 – 5 – 6 2

| / \ | |

7 – 8 – 9 1

For example, a room with 5, 1 would have the model placed in the center on the floor.

* + 1. The user can left click in the room or on a wall (as per the suboption)
    2. Selecting the room will list all models currently in the room with their model type, general location and coordinates.
    3. Only one item can be set at any given unique location/coordinate combination.
    4. Left clicking on the model in the room list will provide the option to delete it, removing it from the room/wall.
  1. Set Event (for map and module only)
     1. Events go before combats if there is a set combat on a location.
     2. Like with Models, events are set with a given room click (identifying the map location x,y), and giving the room 9 point location.
     3. Only one event can be at a given unique room and 9 point location combination.
     4. Prompt the user for event type; explorer, quest, key, battle, treasure, and informational
     5. Prompt the user for event dialog (if not explorer)
     6. Prompt the user for event image (if not explorer)
     7. Prompt for required flag(s) (if left blank, it will always trigger). (if not explorer)
     8. Prompt for flag(s) to remove at start. (if not explorer)
     9. Prompt for flag(s) to add at start. (if not explorer)
        1. Explorer // Walking around visual shifts)
           1. Prompt for type of explorer event; spin, no physical light, no magic light, no light, teleport random, teleport specific (which needs an x, y, z). bright, fog or trap (which needs trap type).
        2. Quest
           1. Submenus for item, flag or question.

Item # (if item)

Flag Name (this is different than 6.3.7 as this is to let the player complete the quest, whereas the other is just to get the message at all). For example, a quest may require the player to kill a boss and return.

Answer field (if question; the question itself is embedded in event dialog).

Flag to add if success

Flag to remove if success

Failure Dialog

* + - 1. Key (block someone from passing if fails)
         1. Submenus for item, flag or question.

Item # (if item)

Flag Name (this is different than 6.3.6 as this is to let the player pass, whereas the other is just to get the message at all).

Answer field (if question; the question itself is embedded in event dialog).

Flag to add if success

Flag to remove if success

Failure Dialog

* + - 1. Battle
         1. Battle will prompt the user for the NPC group listing for that location; NPC type for each group, quantity of NPC in each group; Any special items added for treasure reward.
      2. Treasure
         1. Provide for item # and quantity for 1 or more items.
      3. Informational
         1. Informational will prompt for one-shot or repeatable (all others are one-shot).
  1. Set NPCs (for map only)
     1. Option is only available for maps, not a module.
     2. The user will identify the minimum and maximum range for groups of NPCs player will face.
     3. Menu will list all currently assigned NPC combinations for the level. If none are set the level will auto-populate a set based on the base NPCs for the level and dungeon type when publishing.
     4. The user will be able to add an NPC by selecting from NPCs identified for the dungeon and level range as well as general NPCs.
     5. The user will identify the minimum and maximum quantity of the specific NPC the player will find in a given group.
     6. The user will identify the load type as either random, or the fixed x,y coordinate.
     7. The user can select any NPC in the list and edit the load type, coordinates if fixed, the x,y coordinates, as well as the maximum and minimum # of NPCs. The user can also delete a given NPC from the dungeon list altogether.
  2. Define Quest
     1. 0 or more quest id’s as completed as prerequisite
     2. Prompt for daily or main
     3. Prompt for if it is a raid quest
     4. Prompt for Quest ID#
     5. Prompt for short description
     6. Prompt for dungeon type (if any)
     7. Prompt for dungeon level
     8. Add 1 or more dialog and image combinations
     9. Prompt for flag(s) to remove at start.
     10. Prompt for flag(s) to add at start.
     11. Flag to add if success
     12. Flag to remove if success
     13. Prompt for Quest Type; Kill, Get, Goto, PvP, Construction, or Riddle
     14. ID # (item or npc)
     15. Quantity
     16. Timer
     17. Answer
     18. Location X and Location Y (for Goto)
     19. Reward Type; Gold, Item, Chest, CP, Building
     20. Reward Item #
     21. Reward Quantity
     22. Success Message

1. Publishing a Map, Module or Quest
   1. When the map is done, select Publishing to have it save to a finalized format and location.
      1. This includes generating NPC groups if none are identified.
      2. The map will be saved as normal before saving a finalized copy to the published directory.
      3. Have it randomly place 5 treasure chests around the accessible areas.
      4. Have it warn if there is no stairs up (or down if the level is not 30th.
      5. Have it warn if the walls have any gaps to the outside (map must be fully enclosed).
   2. When the module is done, select publish to have it save to a finalized format and location.
      1. Have it warn if the walls have any gaps to the outside. However, gaps can occur if there is a connection point in the gap location.
      2. Have it warn if there is not at least 1 connection point in the module.
   3. When a quest is done, select publish to have it save to a finalized format and location.
      1. It will validate all required information is given.
2. Map Generation
   1. Prompt the user for the number of maps.
   2. Prompt the user for dungeon type.
   3. Prompt the user for dungeon level.
   4. Prompt the user for complexity by identifying how many modules to use.
      1. It should warn if the number is greater than the number of modules currently made for that dungeon type and level.
   5. Use the dungeon type and level with a counter for the file name as each is created.
   6. Using the connection points, create base maps by snapping the given number of modules together like a puzzle. Each module should be unique (for that map).
      1. Ensure that none of the map module pieces overlap.
   7. Apply up and down stairways in random locations (no down if the level is 30th).
   8. Apply 5 random treasure chests
   9. Apply dead ends to any empty connection points.