RTR Editorv0.1 for Release 1b

# Introduction

Rise to Ruins (RTR) is a game where this editor will modify preexisting save files. The game keeps two types of saves in separate compressed folder with ‘yml’ type files, one array file for a map, and a bitmap. One saves folder is the overworld and the others are sections in this world. There are some links between them and will need to be accounted for. ‘yml’ files hold variables quite messily.

The editor will read in a single save for basic CRUD operations for most files. Not all files will be used but must be saved with them. The editor will handle the ‘yml’ is an XML or JSON format, whichever loads smother. The order of variables, “version” and “lastID” are unsure if required and should be kept as possibly close. Names of files must always be the same. The UI will differ from file to file but can be broken down to a grid, toggle list and item assembly layouts in a tabular fashion.

# Save File Stream

Prompt user to select a directory which holds two folders. One for the World and another for the collection of regions, both holding compressed files of mostly ‘yml’. Backup the directory in resources, decompress the file and create a Game object that’s passed to the Console. Files don’t need to be in stream till selected in the Console.

# 1.1 Console

Functionality will be limited and only a few tabs will show for now. Tabs for all functionality will show as Active/Disable/Error (if file isn’t there or can’t be read). The file should be back-up in case of errors and be locked during editing. The Game object should be passed into the Console and load with what’s working when read. Review layout 1.0 for the GUI.

# Tabs

Each tab will handle a different type of edit to the save. The user should be able to select and do back and forth between tabs.

# 2.1 Mob

There would be a selective list of each mob with their type (Villager or Monster), name, and ID. Upon selection there should be a form that propagates with existing data that can then be edited. Review the mob.yml file for fields. Not all fields will be made editable for simplification. Creation of a new mob should be appended at the end of the list with the next order of mob ID and “mobID”. The overworld files may need to be edited for reflect these changes. A skeleton template may be provided.

# Mob - quick revival

Have a button that will revive all villagers if the dead field exists.

# 2.2 Corruption

A grid should display the pointed [x,y] from the file corruption.yml. These points must be toggleable to be purple /green. All if any purple points exist should be written to the file fields. There can be a cluster of points and should be kept together and not mixed with other clusters and fields. If blank, leave field blank.

# 2.3 Trade

A list of items should display. This is the trade.yml file. Each item will have two fields, ‘buy’ and ‘sell’. These fields can be edited. There can be no duplication and the item must be in the database if a new one is added.

# 2.4 Perk

The perk.yml file in the overworld fold contains a list of different perks. Each perk may be toggled ‘true’ or ‘false’. No perk may be removed or added.

# 2.4.1 Perk - Quick toggle

Button to toggle all perks to ‘true’ or ‘false’.

# 2.5 Chest

The chest.yml in the overworld folder will hold a number of chest objects that can be zero. Basic CRUD operations. ‘chestXP’ is a separate field in the file that can be edited.