ScriptRPG

# Introduction

ScriptRPG is meant to be an educational/RPG game that teaches coding through interactive RPG elements of a game. Every action in the game is made through “code” being ran to make that action happen. Everything from moving, to combat, to chatting with NPCs will have to be controlled through means of coding.

ScriptRPG will randomly generate its terrain, towns, and dungeons. The player will start in a peaceful place with nothing but simple instructions on how to play. The player gets a REPL, or an interpreter line, to put in “code”. A line of code entered will result in an action being done in the world. As the player plays through the game, they will unlock more coding features, such as extra libraries, or utilities, such as scripts to hold code to save and replay code easier. These libraries could include world sensing (locate items and utilities around the player), monster sensing (determine nearby monster names and stats), algorithms and containers (such as dynamic arrays), or even utility libraries (these include key bind functionality).

With the possible addition of skills, players will find that certain tasks are repetitive, and some require strategy. Certain skills, such as smithing, can have repetitive actions, requiring the player to create code to make this task easier. Other tasks, such as combat, will tend to be more strategic. Monsters may have randomized behavior, causing players to devise algorithms to fight monsters. The addition of skills will make ScriptRPG a fun experience, as the player can choose what they want their experience to be. Players can choose to be a notable crafter, who spends their time crafting jewelry, while others can slay the grandest monsters around.

Because of the possible addition of skills, players are not subject to certain classes, as much RPGs restrict. This allows players to be any kind of person they’d like to be. Not only do players have skills, but they also will have stats, such as health, stamina, and even special stats related to coding. Just like how most RPGs have the standard stats (health, stamina, strength, etc.), the player can have stats that effect the execution of their code. This includes execution speed, memory size, and script length. As the player advances through the game, they can choose how to improve these abilities, if they so choose. This also makes players take care of how they code, as they cannot have unlimited resources.

As the player advances through the game, they will find themselves automatically many aspects of the game to their liking. They might be able to finally craft a certain blade without intervention, or even clear entire dungeons without human interaction. There will be no limit at what the player can automate.

# Screens

The main screen will contain:

* Interpreter window
* World window
* Chat Window
* Stats/Inventory window
* Minimap Window

## Interpreter Window

The interpreter window will be where the player can enter in single lines of code to interact with the world. As the player types in code, they will appear a “history” section in the interpreter. This simple interface only contains very minimal stuff.

## World Window

The World Window contains the view into the world of the player. The world will be a 2D randomly generated world. The camera will be focused on the player and the player will always be in the middle of the World Window. All physical objects in the world will appear in the World Window. This includes the terrain, monsters, buildings, containers, and NPCs. Items in the world will appear in either containers or inside a self-made “container” if they are dropped. All entities will appear as sprites in the World Window.

## Chat Window

The Chat Window is a place where important messages are placed to inform the player. This window will also contain several channels for which to filter certain messages. This is also where the player can see any printed-out text from their code.

Messages that could appear:

* Game Messages
* Player-object Interaction Messages
* Code Output
* Player Chat
* NPC Chat

## Stats/Inventory Window

The Stats/Inventory Window will contain several tabs that show the status of the player. This will include current inventory, stats, skills, and settings. Pretty much anything visible not in the Interpreter Window, Chat Window, Minimap Window, or World Window will end up being a tab in the Stats/Inventory Window.

## Minimap Window

The Minimap Window will show entities and buildings that are surrounding the player. This rough view of the surrounding area will have a more zoomed out view than the World Window.

## Windows Preview

# What I Plan to Accomplish in This Class

The world will be very small, is randomly generated terrain is too difficult.

Sprites will be very simple objects, if sprite creation is too complex.

“Code” interpreted will just be strings that are interpreted, rather than a full-blown programming language. This will restrict the complexity of the scripts and gameplay quite a bit.State Diagram

start()

**Loading**

**Title Screen**

# Skills/Inventory Window

The skills/inventory window will consist of a row of button on top and bottom, enclosing an html canvas. These buttons are html buttons that will change the enclosed canvas upon click. The current buttons include:

* Skills
* Inventory
* Equipment
* Settings
* Scripts
* Stats