

Project Proposal

Samuel Xiang

0.1 Introduction

The language I would like to design would solve the "problem" of designing turn-based RPG games on a computer. This language would be able to generate a playable game with an interactable world and combat between characters, while giving the programmer plenty of flexibility to add to the existing structure as they please. In essence, it would provide the programmer with an outlet that allows them to express complex creative ideas in simple ways.

There are a vast selection of unique turn-based RPG's in the real world, but the genre has rules and restrictions of form that are almost impossible to break. As such, expressing the basic tenets of any turn-based RPG can be boiled down to several patterns, which mean that a formal grammar is perfectly capable of encapsulating these tenets. However, building these core ideas over and over again is redundant and time consuming, so a programming language should be a fun way to create these games with minimal labor.

0.2 Design Principles

This language should express lengthy ideas and systems of turn-based RPG games in simple English. From the technical perspective, this program should generate a playable game to the users specifications within the scope of the language. By playable, I mean a game that an outsider user interact with and complete without knowledge of the inner workings - like an actual video game. Because these ideas are expressable in simple English, I would like to create a readable program without lots of weird characters that almost anyone could access and easily intuit. Because the creation of a game involves the creation of a fictional world, the program should almost read as a story or encyclopedia of this world.

0.3 Examples

My language expects a text file of the .rpg format. Example: dotnet run "gamename.rpg"
Example 1.

There is a room named "Foyer" with objects: Playable character named "Samuel" with stats: hp = 10, mp = 10, atk = 10, def = 10, matk = 10, mdef = 10, spd = 10 and abilities:, Nonplayable character named "Katerina" with stats: hp = 10, mp = 10, atk = 10, def = 10, matk = 10, mdef = 10, spd = 10 and abilities:, with description: "A modest foyer with not much to look at. Another person stands in the room.", and connections: North is "Living Room",

There is a room named "Living Room" with objects: Nonplayable character named "Owen" with stats: hp = 10, mp = 10, atk = 10, def = 10, matk = 10, mdef = 10, spd = 10 and abilities:, with description: "The living room is cozy.", and connections: South is "Lobby".

Expected Output:

A modest foyer with not much to look at. Another person stands in the room. ğ

Example 2.

There is a room named "Castle Gate" with objects: Playable character named "Ser Samantha" with stats: hp = 100, mp = 100, atk = 50, def = 50, matk = 25, mdef = 50, spd = 75 and abilities: ability named "Heroic Slash" has effect "deal 80 atk dmg, cost 25 mp", ability named "Warhorn" has effect "buff def 15, buff mdef 15, buff spd 15, cost 50 mp", with description: "The castle has been taken over by a group of marauders! Retake your kingdom at any cost!", and connections: North is "Courtyard".

There is a room named "Courtyard" with objects: Nonplayable character named "Raider Cliff" with stats: hp = 50, mp = 0, atk = 50, def = 25, matk = 0, mdef = 0, spd = 50 and abilities: Nonplayable character named "Raider Shaman

Lernie" with stats: hp = 25, mp = 100, atk = 0, def = 25, matk = 75, mdef = 50, spd = 40 and abilities: ability named "Fireball" has effect "deal 75 matk dmg, cost 25 mp", ability named "Wind Blast" has effect "deal 50 matk dmg, debuff mdef 20, cost 50 mp", ability named "Icefall" has effect "deal 75 matk dmg, stun 1 turn, 40 mp", with description: "The courtyard is littered with the remains of a great battle.", and connections: North is "Throne Room", East is "Training Room", West is "Library", South is "Castle Gate".

There is a room named "Throne Room" with objects: Nonplayable character named "Raider Chieftan Karla" with stats: hp = 100, mp = 50, atk = 75, def = 50, matk = 50, mdef = 50, spd = 75 and abilities: ability named "Rage" has effect "heal 25, buff 25 atk, buff 25 def, cost 25 mp", ability named "Pulverize" has effect "deal 100 atk dmg, cost 30 mp", Object named "Royal Scepter" with description: "The royal symbol of your kingdom." and interactions: "pickup" has effect "end game", with description: "The ornate throne room is marred by the aftermath of the royal guard's last stand.", and connections: South is "Courtyard".

There is a room named "Training Room" with objects: Nonplayable character named "Weapons Master" with stats: hp = 75, mp = 0, atk = 100, def = 75, matk = 0, mdef = 0, spd = 100 and abilities: ability named "Parry" has effect "buff 25 def, buff 25 atk", Object named "Broadsword" with description: "" and interactions: "equip" has effect "increase 25 atk", with description: "The training room, usually stockpiled with weapons, has been looted by the raiders.", and connections: West is "Courtyard".

There is a room named "Library" with objects: nonplayable character named "The Librarian" with stats: hp = 50, mp = 150, atk = 25, def = 50, matk = 100, mdef = 100, spd = 10 and abilities: ability named "Flood Mind" has effect "deal 100 matk dmg, stun 1 turn, cost 75 mp", ability named "Electrocute" has effect "deal 50 matk dmg, debuff 25 mp, cost 25 mp", with description: "The quiet library's books remain untouched by the attackers.b", and connections: East is "courtyard".

Expected Output:

The castle has been taken over by a group of marauders! Retake your kingdom at any cost! ;

Example 3.

There is a room named "Principle's Office" with objects: Nonplayable character named "The Principle" with stats: hp = 300, mp = 300, atk = 150, def = 50, matk = 100, mdef = 50, spd = 10 and abilities: ability named "Expel" has effect "end game", Object named "Principle's Door" with description: "A stately, forboding wooden wall." and interactions: "unlock" has effect "locked by 'Principle's Key'" with description: "A room of misery, dread, and control. This place is a monument to tyranny.", and connections: East is "Administration".

There is a room named "Administration" with objects: Object named "Secretary Pass" with description: "" and interactions: "pick up" has effect "give to character", with description: "A room well-known for frequent visitors of doctor offices.", and connections: North is "Cafeteria", East is "Hallway", West is "Principle's Office".

There is a room named "Cafeteria" with objects: Object named "School Lunch" with description: "A cardboard pizza and some stale fries." and interactions: "eat" has effect "buff 200 atk, buff 200 matk, buff 100 spd", Object named "Janitor's Door" with description: "The grimy door only used by the old, wizened janitor." and interactions: "unlock" has effect "locked by 'Janitor's Key'" with description: "A room of respite and rejuvenation, no matter how disgusting the food.", and connections: West is "Janitor's Closet", South is "Adminsitration".

There is a room named "Hallway" with objects: Object named "Janitor's Key" with description: "An unmarked key covered by a mysterious gunk." and interactions: "pick up" has effect "give to character", with description: "A cage of metal lockers and the odor of highschoolers.", and connections: West is "Administration", South is "Classroom".

There is a room named "Classroom" with objects: Playable character named "The Student" with stats: hp = 100, mp = 200, atk = 50, def = 50, matk = 100, mdef = 50, spd = 40 and abilities: ability named "Information Dump" has effect "deal 100 matk dmg, cost 50 mp", with description: "A place of dozing off and doodling in notebooks.", and connections: North is "Hallway".

There is a room named "Janitor's Closet" with objects: Object named "Principle's Key" with description: "" and interactions: "pick up" has effect "give to character", with description: "Surprisingly clean, considering how unkempt the janitor appears.", and connections: East is "Cafeteria".

Expected Output:

A place of dozing off and doodling in notebooks. ı

0.4 Language Concepts

The language as I envision it right now doesn't actually have many moving parts - the set of primitives I want combine nicely, and the most of the functionality/computation is hidden from the programmer. As such, the programmer simply needs ideas to fill out the game. Primitives include type, name, stats, ability, and effect. Using these building blocks, we can create characters, rooms, and objects, which is essentially all that we need to build the game and the world.

Creating the map, executing combat, etc. are all things that the programmer doesn't consider - rather, they create the guidelines that the functionality of the language operates within. The most important things the programmer has to understand are how rooms are connected and how stats interact. Characters are another core part of the language and the game, in that they will provide a lot of the meat of combat and the game's story. Most of the computation hidden from the user comes in the form of the effect type, which basically dictates the interaction of stats and objects.

0.5 Formal Syntax/Grammar

```

<paragraph> ::= <sentence>
              | <sentence><paragraph>
<sentence>  ::= <character>
<character> ::= <type> <name> with stats: <stats> and abilities: <abilities>.
<type>      ::= playable
              | nonplayable
<name>       ::= named <string>
<descriptor> ::= <string>
<string>     ::= *a string in F#*
<stats>      ::= hp = <int>, mp = <int>, atk = <int>, def = <int>, matk = <int>,
              mdef = <int>, spd = <int>
<int>        ::= *an int in F#*
<abilities>  ::= <ability>,
              | <ability>, <abilities>
<ability>    ::= ability <name> <effect>
<effect>     ::= has effect <string>

```

Syntax	Abstract Syntax	Type	Prec/Assoc	Meaning
<Sentence>(one or more)	Paragraph of Sentence list	Sentence list	n/a	Paragraph is a list of one or more sentences.
There is <type>-character named <name>with stats: <stats>and abilities: <abilities>.	Character of record	Sentence	n/a	Character of record takes in four types and combines them in a record. The primitive types are Type, Name, Stats, Abilities. If these types are not present, an error will be thrown.
<type>	Type of string	string	n/a	<type>is a primitive, represented by a standard F# string.
<name>	Name of string	string	n/a	<name>is a primitive, represented by a standard F# string.
hp = <int>, mp = <int>, atk = <int>, def = <int>, matk = <int>, mdef = <int>, spd = <int>	Stats of record	record	n/a	<stats>is a primitive, represented by a record of ints.
<Ability>(z eroor more)	Abilities of Ability list	Ability list	n/a	Abilities is a list of zero or more sentences.
ability <name>has effect <effect>	Ability of record	record	n/a	Ability is a record of Name and Effect. If these types are not present, an error will be thrown.
<effect>	Effect of string	string	n/a	<effect>is a primitive, represented by a standard F# string.

0.6 Semantics

0.7 Remaining Work