

A More Detailed Plan For TAGml

by David N. Robinson

Description of TAGml

TAGml is both a game engine and a mark-up engine for the creation of text-based adventure games. The engine will provide gaming mechanics (such as an inventory system, a map system, character creation, etc) that will take care of a majority of the technical details required for an adventure game to be created. The mark-up language will allow a game designer to make description of a world map, objects, and non-playable characters (NPCs) on top of the engine to create a full experience.

User Stories

A **game designer** will interface with the engine by writing TAGml files that will be fed into an interpreter through a command line interface. The interpreter will read all the TAGml file(s) through a DOM XML parser, any syntax errors will be returned to the user via **stdout** and the program will exit. The interpreter will also have debugging options to show any information gathered from the TAGml files (such as NPCs, objects, and maps created). This will allow designers to debug different aspects of their game without having to play through it.

The **players** will interface by feeding an zip archive containing TAGml files describing the game world. Assuming no errors were found, the interpreter will open an interactive prompt. From this prompt, the user will receive information about and interact with the game world. Information will be delivered as text blobs printed to the prompt. The user can interact by inputing commands that consist of one or more nouns and a verb. The nouns will be objects/NPCs that the player can see and interact with. The verbs will be actions to be performed upon the nouns. A common command might be **open door**.

Technologies and Processes

TAGml is being developed in Go, a language developed by Google. Go gives the advantage of compiling native code for most common architectures and operating systems. Since the program will already be interpreting the game world through the mark-up language, native code will hopefully ensure that interpreter will operate at a reasonable speed. What is more, Go is a language that is quite easy to deploy to the web, allowing for an easy transition of games from a terminal to a web app.

For source control, this project will be using Git, with a remote repository hosted by Github. Github is a popular social coding platform that provides free cloud-hosted Git repositories. Git is the source control technology that I am most comfortable with, which will allow me to get started planning and coding sooner than I would be able to while trying to learn a new version control system. In addition to code, all planning and documentation will also be stored in the Git repository.

The software process I plan on using is Agile with short, one week iterations. At the beginning of each iteration (Sunday), I will decide my goals for the coming week and assign a difficulty rating to each task. These ratings will not correlate to any real-world time (such as one point for one hour), but are relative. My guess is that I will have to be lenient in the beginning of the project, as I will be getting a feeling of how long different tasks will take me. I plan on using BugZilla with an Agile plug-in to track different tasks that need to be completed. I am deploying BugZilla on an AWS instance to ensure that my bug tracker will be available from anywhere.

In an Agile process, daily team meetings are a staple of team building and project management. Currently I am working on this project alone, so I am going to keep a daily log of my activities. This log will be stored within the git repository.

Timeline

This project seeks to create a mark-up language as well as an engine. As a result, the first goal is to design a language specification as well as a game system description.

Week 1

- Write TAGml game system description (12 difficulty)
 - What rules will dictate the game world
 - How can the designers interact with these rules

Week 2

- Write TAGml mark-up language specification (6 difficulty)
- Write TAGml mark-up language XML-Schema (3 difficulty)
- Write a “Hello World” TAGml document (2 difficulty)

Week 3 and Beyond

TBD