Proposal/FAQ For TAGml

by David N. Robinson

What is TAGml?

TAGml (Text-Based Adventure Markup Language) seeks to make text-based adventure game design considerably easier. Rather than coding the specific logic of a game, a basic game logic will be provided through TAGml engine. The engine will provide the following features and more:

- A simple description language that allows game makers to focus on storyline and creation rather than simple mechanics
- Support for user defined actions (verbs) and objects (nouns)
- Built-in inventory system
- Built-in health-meter
- Both a command-line interpreter and an HTML5 web interpreter

Why should I care?

Old-school adventure games are incredible fun that encourage creativity on the part of the creator as well as the player. This project hopes to make the creation of such games less technically cumbersome so users can focus on creating awesome worlds rather than the technical details.

How would the engine work?

The engine is manifested in a command-line and a web interpreter. Both interpreters will provide essentially the same experience, so I am going to speak of them both as the interpreter. One would feed the interpreter a folder containing TAGml files. The interpreter would construct the game data structures from these files and begin the game-loop. The game-loop will prompt the user for instructions (usually consisting of a verb and/or a noun). The interpreter will search its dictionary of terms to find a valid match/perform the proper action. These actions will have in-game consequences, such as moving the protagonist to the North or opening a treasure chest. Once an action is complete, a string describing the result will be printed to the screen and the player will be able to perform another action.

Got a time-line?

Right now, it is fairly lose. All I know is that there will be a final release of this project in early December 2013.

How do you plan on making this?

I plan on using a variant of Agile software development. Since a large portion of this project will be designing a language specification (most likely a XML schema), planning and design need to be a central focus of the development process. Traditionally, Agile values working software over a plan. In this case, the schema (a plan in its own right) is an essential part of the project. To accommodate this, I am shortening the iteration period to 1 week and allowing for user stories to consist solely of fleshing out a part of the language schema.

In addition to the style of software development, this projects hopes to utilize multiple software stacks. The final choice in stack will occur during the first iteration, but for now, Google's Go, Javascript, and Python/Django are all being considered.

Who else has done this?

This engine gets its inspiration from retro text-based adventure games such as adventure and Zork (both found in the `bsdgames` package in Debian based Linux distrobutions). In addition, many of the mechanics of how the game would work come from traditional table-top RPGs such as Dungeons and Dragons (commonly referred to as D&D).

Want to get involved?

Feel free to email me at `darobinson at clarku dot edu` if you want to get involved. This is an ambitious project with a tight time-line. Any and all help is most welcome.