

# Programming Languages & Translators

## Whitepaper for "GameWizard"

——Online Tabletop Games Design Language for Programmers

Team Member: Liyuan Zheng(lz2375)

Chaozhong Lian(cl3190)

Yue Huang(yh2640)

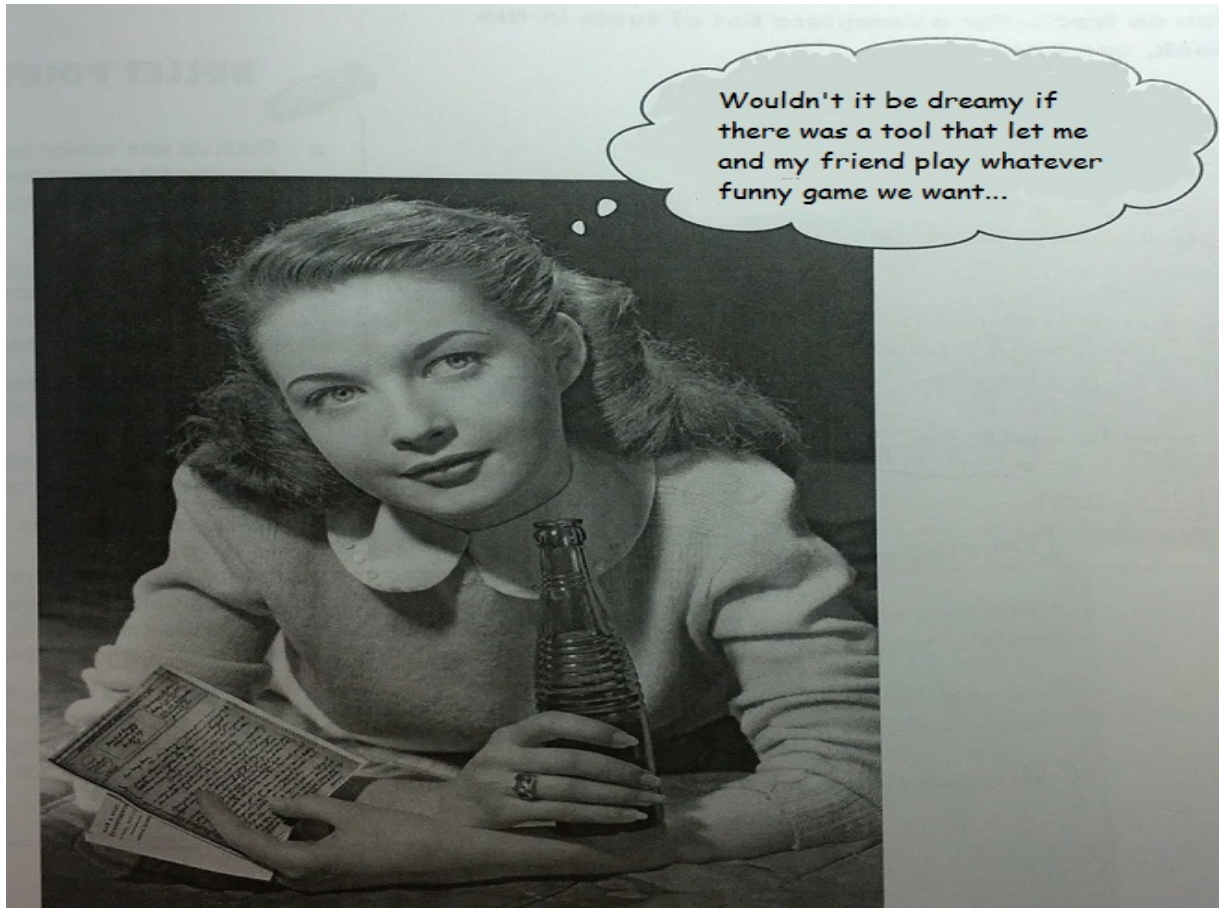
Ke Liao(kl2735)

Songqiao Su(ss4555)



## 1. ***Yes! Playing any game you want with your friends on-line!***

As programmers, we sometimes enjoy creating tabletop games and play with friends. However, almost everyone of us has complaint that getting a group of friends sitting down together in the same place at the same time is insanely difficult, let alone the problems of lack of playing pieces.



And here is the savior of the day! The idea of GameWizard comes from the fact that we are all experiencing such a disappointment when we could not start some kind of really enjoyable games with friends, simply because the small problems like lacking playing pieces. With GameWizard, people can now easily start a tabletop game, whether the game is common or not, without worrying about playing pieces, or that folks can not get together.

With the help of GameWizard, all that we need is some game rules and some leisure time. You can define whatever turn-based tabletop games you and your friend are eager to play. It should be super easy to write normal card games (poker games, Uno, etc.) with our language. Moreover, to satisfy our target, the programmers, many easy-to-use features would be provided to create complicated games, role-playing card games for example. Internally designed with an on-line platform to support remote interactions by different players from different hosts, we shall help a lot for the friends-getting-together problem.

GameWizard abstracts out the common patterns of turn-based tabletop games, making the language shorter and most efficient than simply writing java or python. Using

GameWizard to define new games would be a simple and interesting process, so why not start to be creative?

## 2. Why our language is so different?

➤ Define cool games, like RPG CARD GAMES, with less lines of code.

Our language, GameWizard, aims to provide an easy way to design complicated turn-based tabletop games( like Magic The Gathering, HeartStone, etc.). By abstracting out the common operation, attributes, and state, we simplify our language so that programmer can write much less codes than writing in a general-purposed language, like Java. Our language also simplifies the initialization of List, HashTable like container, further simplifying the code. For example, our language will integrate keywords like “shuffle” or “deal-card”, shortening the codes for very complicated



role-playing card games.

Every turn-based tabletop game has three parses, the starting parse(initializing the count of "blood" and "mana" for each player, setting up board games, distribute pokers for each player, etc. ), the middle parse (this is the parse when every player performs some actions in turn, using some card, invoking some effects, etc.), the ending parse(deciding which player is the winner). Also, commonly in role-playing games, every player has some abilities (be it releasing monsters or creating fire) or may have some magical items (something like magical ruby or magic wand). Our programming language provides simple interfaces for users to define their fancy games easily.

➤ **Compiles, and yes, now play on-line!**

Once the game is written, all that users need to do is to simply compile the source code. After that, a server program and a client program will be generated separately at

once. The user now only need to distribute the client program to his/her friends, and run the server program. All his/her friends can now connect to him/her from the client programs. And whoa la, the game works like a charm!

### **3. Who are the “programmers”? Why is the language good for them?**

As of today, many online platforms are already there for the common tabletop games like poker games and Chess. For those games, learning a new language can hardly be worthy. But the speed in the industry world can never catch up with the creativity of our programmers! We always have good ideas of games, but java debugging can be depressing. Do I need to mention those brilliant, but only-in-C hardware engineers?

By “programmers” we do not mean experienced software engineers. However, as being able to create fancy and interesting games, we assume them having an idea of programming, being able to make a list of components of games ( players, cards and attributes of them, etc. ) and use for loop, if-else statement to represent logic. And that is it.

Simplicity is never a thing that programmers refuse! As we have mentioned previously, our language is fully targeted, we optimize many keywords and syntax intended to specify the type of games, which makes codes much less than simply writing Java code. Besides, our program internally builds the server program and the client program, so our programmers can save their time from tackling networking stuff for online multiplayer games, and focus on the real game designing.