

Software Engineering and Large System Design
Spring 2017
Project Description

This semester, each group will implement its own complete application allowing users to play the game of SET against each other over the Internet. You can download the official rules (in various languages) from <http://www.setgame.com/learn-to-play> (this page includes links to rules of several games; we are only interested in SET). We will also discuss the rules in class. More information about SET can also be found at <http://www.setgame.com/set>. You can also purchase the game from this site if you wish (or from Amazon or several other websites).

A few pieces of information that the rules at the website do not make entirely clear:

- Since there are four attributes, each with three possible values, the total number of possible cards is $3^4 = 81$. Every such card exists exactly once in the deck. There are no duplicate cards.
- If there is no set on the board, but cards remain in the deck, three more cards are dealt face up on the table. This continues until one or more set is possible. (It is very rare to need more than 15 cards visible, and I have never seen it necessary to have more than 18 cards visible, but I have read that it is possible to have up to 20 cards visible with no set. Since three cards are dealt at a time, this means there may be as many as 21 cards visible at one time.)
- You do not deal out new cards unless there are fewer than 12 cards on the table.
- The game ends when all of the remaining cards in the deck are on the table and there is no possible set. It does not matter how many cards remain.

Note that in the real game, if a player declares that they have found a set but they are wrong, they lose a point. You do not necessarily need to implement this feature. You may, for example, have users check and uncheck cards, and unless there are exactly three checked cards which represent a set, anything else they do will go unnoticed. Alternatively, you may require the user to click on a special button in order to submit an indicated set, in which case you might decide to have your system deduct one point if it is not correct. Also, in the real game, new cards are not dealt unless every player agrees that there is no set present on the current board. I think for the electronic game, it makes more sense for the server to notice this instantly and deal out more cards; then players will know that there is always at least one set on the current board. The system should also detect the end of a game and notify all of the players in some appropriate manner.

Each group's application should contain at least the following components:

- A web-based registration system that allows users to register. Users must provide at least a username and a password; you can decide what other information to keep track of. I suggest using MySQL as your database management system.
- A user-friendly graphical user interface (GUI) that allows users to request games or respond to requested games, and then to play against other users in SET. If you wish, you may implement a lobby where users can create and join games. Ideally, your system should be able to handle multiple games with more than two players each, at the same time. Minimally, your system must handle multiple two-player games or one game with several players.
- A server that handles the communication between GUI clients involved in games of Set. The server is responsible for recognizing when a player has indicated a correct set, and for informing each GUI involved in the game of the updated scores and the updated board.

Every group will have their own virtual machine, provided by VirtualBox, running on a server at Cooper Union. A semi-recent version of Ubuntu will be installed on each virtual machine. You will be able to ssh into the machine, and optionally you can run a vnc server to allow remote desktop access. (I have found that the default desktop sharing feature is not secure.) I suggest that each student sets up their own development machine to work on the project, but the final version of the server should reside on your group's virtual machine. (The GUI should be downloadable, and it should run correctly on any Linux or Windows machine.)

The project should be implemented in Java. I recommend using Swing for the GUI, MySQL for the database management system, and JavaServer Pages (JSP) for the web-based registration system. Each group can decide on its own how to handle the client-server communication. Many of the tools and resources that you will be using for the project will be covered in class, but you will need to explore them in further detail, and you may need to learn about others on your own.

The server will need to recognize when no set is present on the board and whether or not a player's declared set is valid. I think it makes sense to have the server detect all sets that are present using the current cards. I recommend creating a simple way to indicate a current board configuration to the server and having it list all of the sets that it finds. This would be helpful to test and debug the set-finding functionality of the server.

There is a lot of flexibility in this assignment. You need to decide how much of the game logic is to be understood by the client (I recommend placing most, if not all, of the game logic on the server). You need to decide how players submit sets and what happens to the other players' interfaces when this occurs. Even the specifications listed here might be open to change, but you should talk to me if you are thinking about this. For example, if you want to use a free database management system other than MySQL, or if you want to use a programming language other than Java, I might be OK with changes such as these, but talk to me about them first. The basic requirements, though, should not change. Optionally, you may decide to also allow users to play Set in solitaire mode, to keep track of players' statistics over time, to implement chat functionality, or to implement other additional features.

All projects will be due at the end of the semester. Each group will give a presentation to the class describing their implementation decisions. During the final exam slot for the class, the final systems will be demoed, and the class will play games using each of the working systems. I will also ask for short individual write-ups specifying who did what on the project. Project grades will be determined mostly based on the final state of the project, including: the user friendliness of the game's GUI (e.g., how easy is it to submit a set and how smooth is it when other players find a set?); the user friendliness of the registration system (e.g., is it easy to create new accounts and update accounts?); the robustness of the system (e.g., are there any bugs?); the speed of the system (e.g., is there any lag?); the features implemented (e.g., did you implement any optional features?); etc. If I feel that the work was distributed fairly and that everyone fulfilled their responsibilities, I will assign the same project grade to all of the members of each group. However, if any student does not do their fair share, forcing other students to do extra work, I might assign different grades to students within a group. I strongly advise all groups to start early and work consistently throughout the semester; also have fun with it!