**Introduction**

You have been hired by a startup software development company, KnowSys. At this company you will work as part of a small software development team to develop a game. The goal of the game is to compete in the marketplace with the well-known game of Risk, a Parker Brothers' game. Parker Brothers is a division of Hasbro. For those of you who are unfamiliar with Risk, the rules for the original game can be found here:

http://www.hasbro.com/common/instruct/risk.pdf

KnowSys is developing its Risk-beater game which it is called *CollegeConquest*.

**The game**

Each player in the game starts out as the Chair of a department. The goal is to advance through the role of Dean of a school, to eventually become President of the university.

A player becomes Chair of a department when they control all the students in that department.

A player becomes Dean of a school when they are Chair of all the departments in the school.

A player becomes President of the university, thereby winning the game, when they are Dean of each school at the university.

There are six schools at the university, with the indicated departments:

Arts
- Music, Theatre, Dance, Visual Arts

Sciences
- Biology, Chemistry, Physics, Mathematics

Humanities
- History, Linguistics, Literature, Religion

Engineering
- Computer, Electrical, Mechanical, Civil

Law
- Business, Family, Criminal, Immigration

Medicine
- Anesthesia, Otolaryngology, Pathology, Pediatrics

Students can be moved from one department to another at the direction of their chair. Students remain loyal to their chair, even if they are moved into a department chaired by another player.

Students can be moved from one department to another only if they are connected. All departments within a school are connected. Other connections are as follows:

Music <> Physics
Visual Arts <> Mechanical
Linguistics <> Computer
Religion <> Mathematics
Chemistry <> Anesthesia
Civil <> Business
Criminal <> Pathology

Deans battle for control of departments by convincing students to elect them chair. During an election each student in a department votes; students vote for their own chair with a probability of 0.66.

Thus, when an election is called students in a department are polled, and they vote for secession to a new school with probability 0.66 percent if they are students in the new school, and against it with a probability of 0.34 otherwise.

**Game set up**
This is a 2-20 player game. Each player starts as a department chair, and is assigned a single department at random. Thus at the start of the game there will be some departments without a chair. Each player gets 3 students to place in their department.

**Gameplay**
At the start of each turn players get extra students:
- for each department of which the player is only chair: 3 students
- for each department of which they player is also dean: 5 students

In addition,
- for each school of which the player is also dean: 10 students

These students can be placed in any of the player's departments.

The player can then move their own students between adjacent departments, and only one 'hop' at a time. Thus students can move from Music to Physics in one turn, but two turns would be needed to get students to move from Music to Chemistry.

At least one of a player's own students must be left in each department for which they are chair.

A player may initiate an election in any department in which they are not chair but in which they have students. When a player becomes chair of a department through election all students in that department become loyal to the new chair.

When a player becomes dean of a school all students in that school become loyal to the new dean.

**User stories**
User stories, descriptions of required functionality, will be given to each team each week in recitation. Each team is expected to work on the new user stories (in addition to any unfinished user stories from previous weeks) in the coming week. At the end of each week teams must submit their project in its current state. This submission must happen by 9:00 PM the evening before the recitation meets.

**Submission**
You will submit via Web-CAT, on or before 9:00 PM the evening before the team's regular recitation. Each team will make ONE submission. Decide early on who will be responsible for making your team's submission.

**Grading**

Your team's FINAL project submission will be graded on its merits as follows:

*Functionality: 60%*
30% is awarded for stage 1 game-logic functionality, and 30% for the user-interface.

*Documentation: 10%*
You must have a brief user's manual which described how to start and how to play your game (including the commands/rules AS THEY WORK IN YOUR GAME). In other words, if your game does not enforce turn-taking, say so!) Name your user's manual User.txt (it must be a plain text file).

You must also have javadoc comments for each class and each public method which says WHAT the class or method does, not HOW it does it. For methods, the role and expected value for each parameter must be explained, and the return value must be documented too (describe the significance of possible return values).

*Testing: 30%*
You are expected to have JUnit tests for the functionality of each NON-UI public method you write. Do not write tests for your UI code. We are still not aiming for perfection at this point, but we do expect to see a good suite of sensible tests.

Your overall grade is based on the overall team grade for the submission, and your average peer grade. Basic point: if you do not contribute as much as others on your team you will not receive as much credit as your teammates. More details about how peer evaluation works will be made available soon.

**Team meetings**
To help you manage your time, here are some do's and don'ts for recitation:
• Do discuss accomplishments (goals met) since last meeting problems (goals not met) that have come up since last meeting goals for next meeting schedule for the next week for pairs to meet to pair program.
• **Do NOT even consider writing code during your recitation time unless you have done the above.** Remember, your recitation time is the only time during the week when you are guaranteed to be able to get together as a team. This time is precious - don't waste it.
• Do have someone take notes each meeting, and keep notes in your code repository. Each team **MUST** submit one set of meeting notes per week. For the first week of stage 1 your repository may not be set up yet. Once repositories are set up, put a copy in your repository, in a folder called "Minutes".
**You must also e-mail me a copy of your team meeting notes. The subject line of the e-mail must be**

[CSE116-T<xxx>] Minutes from meeting on <DATE>

**where <xxx> is the team number, and <DATE> is the date in mm-dd-yyyy format.**
• Do compromise.
• Do NOT exclude team members - use all your team's resources. Each team has people with different skill levels and talents. Coding prowess is just one of many skills needed to complete this project. **When pairing, strong and weak coders should work together, and the weaker coder should drive more often than the stronger coder (to help them build their skills with the help of the stronger coder, who navigates for them).** Doing this will pay off big-time in later stages of the project! Likewise, not doing this will cost you big-time down the road!