CST8218 Game Project Example

This is an example project that your team can use as a guide while working on your own project. The requirements for this project can be used as a guide for what is expected, and how to approach the software development, regardless of your team's specific project.

Requirements

Using the Sprite Case Study demonstrated in class as a starting point, giving a JSF interface priority over a canvas interface, implement a Sprite-based networked multi-player game with the following requirements:

Tiered architecture similar to the architecture of the Address Book example project studied in class (presentation layer with JSF pages and a named managed bean, a business layer with an Enterprise Session bean façade, and a data layer including an SQL database engine and database schema).

Manipulation of Sprites through JSF, including changing color

Data Validation on JSF manipulations

Secure User Registration and Login (SSL not required)

Internationalization (language/locale explicitly selectable by the user, or detected in HTTP requests)

Some use of AJAX

Rest Interface to support future Mobile clients

Game Aspect

You are free to design any game you wish, subject to the requirements. The recommended approach is to add each requirement to the case study code in the most minimal way possible, and after requirements are technically satisfied, then move on to enhancing the result to be a more compelling game. The following questions are meant to stimulate ideas for your game design:

- What is the goal of the game?
    - What governs the creation/destruction of sprites?
    - Eliminate the other players' sprites by shooting at them (the sprites)?
        - Projectiles are another type of sprite
    - Can sprites be captured, or change teams?
- What information is available to the user on the JSF pages (possibly with a canvas)?

- Does something happen when sprites collide?
- How does the game player control the action? Can AJAX aid implementation?
- Can you animate the sprites with an HTML5 canvas?

Recommended Steps

- I recommend that you do your development in a virtual machine that you can take snapshots of, often, at least between steps. If your laptop is slow or for whatever reason you don't want to work with virtual machine snapshots, be sure to make commits to a source code repository often, whether or not you're using virtual machine snapshots.
- Take inventory (including the annotations) of the Sprite Case Study code (you are not required to deal with the Java Swing Client, because the scope of this course is Web Applications, but you might find the Swing Client useful)
    - Understand how the application works, and the role of each java class in the project (**find all of the annotations** in the Java code of the project, especially annotations right above any Java class definition)
    - Draw the architecture diagram with the names of classes, for future reference
- Add authentication to the application using the File Realm, where you add the users manually to the realm using Glassfish, and add a security constraint to require authentication
- Add Data Validation that would prevent the user from making the position of a Sprite negative (for example, prevent positions of x=-4, y=-100)
- Add a converter to convert a Color object to a string for presentation to the user on the JSF page, and accept user changes to that string (giving the user a way to change color). Adding a color-picker is possible, but not required (the converter is required).
- Add a second Localization Bundle with any second language (you can make one up) so that the labels on the JSF page(s) are shown in the user's language
- Enhance the authentication by switching to a jdbcRealm. Add the ability for users to register, with the authentication Realm to do authentication (first BASIC, then Form-based)
- Add a REST interface that can be used to retrieve all sprites. Can your REST interface be used to determine how many sprites?
- Incorporate AJAX in your JSF page (start with one of your JSF pages)
    - Note that the Lecture in Week 12 is about Ajax: plan accordingly
- Add an HTML5 canvas and show the sprites on the canvas
- Think about your desired game design, and bit by bit make changes to bring your app closer to the design, and repeat
    - Ideas
        - Give the user a way to select a sprite
        - User arrow keys to change direction, and speed?
            - Left key -> turn to the left slightly
            - Up key -> increase speed
        - Add the ability for sprites to collide or interact

- Add the ability for sprites to shoot projectiles