Project 2 Pitch

Initially I was incredibly excited at the opportunity to work on a game in Java! As I've stated before, game audio design and integration are career paths I am interested in exploring in the future. While most of that is done in C++ and C# in the modern era, some truly legendary titles have been developed in Java.
For my project I decided to go with a brick breaker clone with a few twists. I'll get into these twists a little later on but I figure I'd start with the basics. As I'd mentioned this is a brick breaker "clone" as the fundamental gameplay isn't something I'm trying to augment too much. For those not familiar brick breaker is a game in which a ball is spawned into play, used to break rows of bricks by bouncing it off a paddle the user has control over. Once all the bricks are broken you win but if the ball misses the paddle at any point and goes off screen the game is over. This is baseline game has no need to be changed, while some developers have added extra gameplay features I wanted to focus on style as apposed to the former. Considering the recent work we've been doing in class, I thought it'd be a good opportunity to create custom textures and implement them into the game. I also have a desire to create custom made sound effects that will be integrated as well. These effects will make up the sounds of the ball hitting the bricks and the paddle. There will also be separate sounds for winning and losing. With these aesthetic features added, I'll also be creating a score counter that will keep track of high scores up to the max possible score (all bricks broken).
All in all this should be a solid project with some interesting challenges to overcome.

Timeline

Weeks 1-2: Design and basic functionally completed
Weeks 3-4: Aesthetic Design and Testing

LO Break Down

This section is a little breakdown of how the learning objectives of the project will be executed within this project.
Right off the bat there are a few LOs that come as an obvious in terms of their implementation within a project like this. Firstly a game is object-oriented in nature in terms of creating multiple objects to handle things like the score, ball, bricks and so on. In the same vein, classes will be used to denote functionality with a focus on MVD and will no doubt require inheritance and polymorphism in their implementation. This in turn covers the LO dealing in GUI design as of course the games display is just that; a GUI.
There are a few LO's that aren't as obviously implemented within this project, for example the use of arrays. My goal is to use an array within a scoring class(or something equivalent) as a means of storing user data such as scores and names. On the same subject, I will create constraints on naming conventions which will need to be upheld via some sort of exception handling. Lastly, I'm not entirely sure how I'll implement textile IO. My initial ideas are to use a text file as a means of storing mock scores for new players to compete against but of course I'd love to implement something a little more involved than that.
Over all I'm confident in the requirements being met within this game and am excited to come to the final product and test it!