**Shadow Chess**

**Detailed Design**

**COP 4331 Spring 2014**

Team Name: Group 20

Team Members:

* Patrick Delva
* Alvaro Velasquez
* Stephen Monn
* Walker McDonnold
* Brad Costa

Modification history:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| v0.0 | 05/13/13 | S. Applegate | Template |
| v1.0 | 03/27/2014 | P.Delva | Initial Release |
| ... |  |  |  |

**Contents of this Document**

Design Issues

Detailed Design Information

Trace of Requirements to Design

Design Issues

In this first prototype we looked to have a working chessboard with sprites that were created by us and display their movements. The chessboard was designed to have tiles be large enough to display at least the top half of sprites in the background so that when we came to using the game itself, the user would not have much difficulty. Sounds were added to accompany the moving of pieces and tiles of different colors displayed the movements that they could make upon a click. We attacked the performance issue by not adding sliding or having each piece stop on multiple tiles as it moved. Basic testability in this stage was nothing but seeing if movements were legal and that sprites followed their displayed paths.

The second prototype concentrates on the use of code to provide turns to players in game, ending the game, and resetting a match. The pieces are finalized with this stage and have full move support outside of transformations. We believe to have a full game of chess working (advanced functions like castling are still in the works).

Our third prototype will include transformed sprites (deciding between borders or new sprites altogether) for the pieces and a secondary moveset for the time with which pieces are changed.

Detailed Design Information

<In this section, you include diagrams, listings, etc. to provide the complete detail of the design of your product. A class diagram is required. Depending on the tools or development environments you are using, you may be able to automatically generate much or all of this section.>

<NOTE: The design documentation must be well organized, modular, and of sufficient detail so that, using only the concept of operations, SRS, and design documentation (and NOT the code):

* the technical reviewer or software maintainer can fully and easily understand the structure, function, and non-behavioral characteristics of the software
* the coder can develop the final product from the design with NO decisions to make other than syntax
* the maintainer can safely determine how to modify or enhance the code and know what other sections of code may be impacted by the change>

Trace of Requirements to Design:

<In this section, provide a trace of each requirement in your SRS to the design: for each requirement, where (in the design) is its implementation?>