Requirements and Specifications

Bryan Ceberio-Lucas Ethan Eldridge Peter LeBlanc Phelan Vendeville

***Abstract*—This document contains the specifications of CS 205 Software Engineering's final project, an implementation of Rat-a-tat Cat. These standards and requirements will be followed by all team members. The following terms and descriptions must be clear to all members so that the system is a cohesive and comprehendable system.**

CONTENTS

I. TERMS AND DEFINITIONS

Must If a specification uses the word Must, it is mandatory that all team members follow this requirement. E.g. *The System **must** handle all possible URLs and direct the user's to an appropriate page.*

Shall If a specification uses the word Shall, then the System must respond to the specification in the detailed way. E.g. *The system **shall** perform operations in a timely manner and no operation will take more than 10 seconds*

Gantt A bar graph used to visualize a project schedule

Glow To glow is to surround an object with a faint highlight that indicates that the User may interact with this object.

State The System's internal state is kept using a Stack of strings that indicate the current and next state of the System, this collection is refered to as the State and can be pushed, popped, and peeked.

II. INTRODUCTION

Phelans awesome introduction goes here

III. INTRODUCTION

IV. SCOPE AND PURPOSE

A. Scope

What is the scope of our project

B. Purpose

This is where the purpose of our project goes

V. FUNCTIONAL REQUIREMENTS

The functional requirements of this project are specified by the Coding Standards in §V-A, Version Control standards in §V-B, directory and game Architecture in §V-C, Artificial Intelligence logical overview in §V-D, and Database Design images and naming conventions in §V-E.

A. Coding Standards

The following standards must be followed by all team members. By defining these standards all code will be readable for all members, and no discrepencies between conventions will occur. Each team member is responsible for keeping to these standards, and submission of code not keeping to these standards will come under review and the format shall be adjusted accordingly.

Naming conventions

- Variable names must be camelcase, descriptive, and self documenting
- Class names must begin with a capital letter and use camelcase
- Database table names must begin with a capital letter and use camelcase
- Database table names should be short, one word where ever possible
- Directories must be lowercase and without spaces
- File names must be lowercase and without spaces
- All images should end in .png and be of that format
- CSS class names must be self-documenting
- CSS class names must be camel case
- Constants in any form must be all uppercase with underscores between natural breaks
- Git tagging must follow the convention of version_x.y, x must be the major release number, y the minor release number
- The team leaders repository should be refered to as mainline during remote declaration

Commenting Conventions

B. Version Control

C. Architecture and Structure

D. Artificial Intelligence

E. Database Design

VI. Non-Functional Requirements

A. User Interface

pretty pictures and descriptions galore

B. Game Play

This is where the storyboarding stuff goes

C. Character Design and Concept Art

D. Timeline and Delivery

This is where timeline and due dates go as well as what has to go into each part

VII. Test Cases

VIII. Summary

Overall summary