**Technical Design Document - CMPS 20 - Game Design Experience**

The purpose of this document is to describe the internal structure of your code at the class level. This is accomplished by creating one or more UML structure diagrams to represent the static structure of the code. These diagrams represent the detailed technical design of your project.

A major goal of the assignment is the application of software design patterns in the context of a non-trivial software design activity. As a result, your design must use design patterns where appropriate. While the assignment encourages the use of software design patterns, and expects their use, gratuitous use of patterns where they are not appropriate is strongly discouraged. A good designer knows when to use software design patterns, and also when not to use them. You need to indicate on your UML diagram when you are using specific software design patterns.

A complete technical design document will contain a static structural description, detailed below.

*Static structural description.* In one or more UML structure diagrams, represent all of the classes and interfaces that your team will develop in your project. Where you create subclasses of existing XNA framework classes, you should also indicate these classes as well on the UML diagram, though not to the same level of detail. Each class should describe the class variables, and all of the methods in the class. Ideally you will also provide method parameter lists, though this is not strictly necessary. Diagrams should represent all inheritance and containment relationships among classes. The structure diagram typically needs to describe the following game elements:

* input handling
* representation of game levels, if present
* player class
* enemy classes, including a container holding active enemies
* game object classes, including a container to hold them
* collision detection
* classes for drawing player, enemies, other game objects
* classes for handling audio
* menu system classes

Your game may also include other elements above and beyond this list.

When developing your UML structure diagram, you should consider the most important code sequences in your game, as these will lead you to flesh out the details of your design. These sequences should include at least the following:

* Initialization. What happens when the game first starts running? This involves creating player and opponent objects, loading levels, playing background music, displaying an initial menu, etc.
* Menu system. IF the game has a series of menus, what is the sequence of selecting one menu item?
* Main game loop. What methods are called, in which order, during each clock tick of the game during normal operation? Typical operations include gathering input, updating state, collision detection, checking for end-of-level conditions, drawing objects, and updating particle systems.
* Player collision. What happens when the player has a collision, such as with a wall, a bullet, an enemy, or other? This involves updating object state, visually representing the collision, and often playing some kind of sound.
* Enemy collision. What happens when the enemy has a collision?
* Scoring. How is the game score changed?
* AI. What are the different possible behaviors of opponents?
* End of level. What happens in the game when there is a transition to a new level?

*Common pitfalls.* Common errors include not including audio handling, and inconsistency between the game design document and the technical design document. Every aspect of the game described in the game design document should be present in the technical design document as well.

**Description**

To aid our understanding of your UML structure diagrams, you must also provide a brief, 1-2 page description of your design. Provide a brief 1-2 sentence textual description of the responsibility of each class in your design. Please also briefly describe any unusual (i.e., beyond getter and setter property) collaborations among classes. This should not be a long document.

Last updated: 1/5/2009

References:

<https://classes.soe.ucsc.edu/cmps020/Winter09/tdd-template.html>

<https://classes.soe.ucsc.edu/cmps020/Winter09/tdd-description.html>

UML Tools

<http://staruml.io/>