Project 3 - Design Document  
IMGD3000  
Brian Seney, Alfredo Porras  
  
Game Name – The Legend of Rogue Ball  
Team – Epic Sports Games Inc.  
Genre – A single player sports game with rogue-like action adventure elements.  
Game Description – The player controls a character that can move around and fight monsters. The player’s goal is to take a ball to a goal while avoiding traps and monsters on the way. The player has several weapons at his disposal, such as a bow and a sword, that can be used when the player is not holding the ball.  The monsters feature basic AI and can stalk the player and attack.  Monsters include a large, hulking monster that is slow and takes up several blocks, a monster that uses a long range attack, and a monster that steals the ball from you.  Obstacles to overcome include breakable boxes, movable blocks, and traps, which add puzzle elements to the otherwise action packed game.    
Technical Features – The game includes a monster class which all monster types derive from, and a ball class, which uses velocity and acceleration to move.  There is a weapon class, with different attacks and behaviors for each weapon, such as a long range attack.  There is a Game Controller class, which holds data about the players, loads levels into the game, and manages game flow.  The game controller loads levels into the game from text files.  The different block types, such as movable, immovable, and breakable, will be objects derived from the GameObject class.  
Artistic Assets – Due to the limits of curses and the Dragonfly engine, most objects are one character.  This allows the levels to be large, relative to the size of the characters.  This also reduces the need for large numbers of artistic assets.  However, there are several main artistic assets present in the game.  The first is the title screen, which is an ascii-art title.  It is three frames, to cycle between colors.  The other art assets are the instructional screens between each level, which are represented as Sprites.  They do not have multiple frames of animation, but are large, and contain text to show the player how to play the game.  
Implementation Plan – The main classes of the game were implemented first, such as the player, weapons, and monsters. The main functionality, which is to move the ball around, was the highest priority, and was implemented first. Once those worked (and the capability to add more weapons and monsters easily were established), other obstacles such as boxes and traps were implemented.  Once the game logic behind the main classes was complete, levels were implemented, along with the capability to switch from one level to the next.  Then, the levels were designed, tested, and loaded into the game.  At that point the alpha was complete, and the game was polished for this final version.  
Distribution of Work –

* Player Class - Brian - Core
* Ball Class - Brian - Core
* Goal Class / Win Condition - Brian - Core
* Monster (base class) - Alfredo - Required
* Multiple Monsters - joint responsibility - Required
* Monster AI - Alfredo - Desired
* Weapon (base class) - Brian - Required
* Multiple Weapons - joint responsibility - Desired
* Block (base class) - Alfredo - Core
* Multiple Blocks - joint responsibility - Required
* Level Class - Alfredo - Required
* Multiple Levels - joint responsibility - Desired
* Art Assets - joint responsibility - Desired

Schedule –

* Monday, February 13 - Game engines combined.
* Monday, February 13 - Begin work on game; basic classes defined.
* Wednesday, February 15 - Submit this document; have one instance of player, monster, weapon created and working.
* Friday, February 17 - Define block classes (movable, immovable, breakable), create more enemy types.
* Tuesday, February 21 - Submit alpha stage.  Begin work on levels, level loading.
* Friday, February 24 - Levels load successfully, implement several levels that can be reached in succession.
* Sunday, February 26 - Title screen, tested game flow, final boss, win condition present.  Game is polished and submitted.