UNIVERSITY OF WASHINGTON BOTHELL

PROJECT PROPOSAL

CSS 450 FINAL PROJECT

Inverse Donkey Kong Tower Defense
Pirates vs Ninjas
(working title)

Jashan Dhaliwal
Harrison Foro

11/15/2010

TABLE OF CONTENTS

INTRODUCTION AND PURPOSE STATEMENT

For our final project, we have decided to create a mash up of two of our favorite genres of games, Tower Defense, and Donkey Kong. Our design will involve the hero characters and the enemy characters spawning at opposite sides of a vertical based map. The hero characters will be deployed to static defense towers, while the enemy characters will attempt to reach objectives the player is charged with defending. Gameplay will be broken up in to several stages, each stage getting progressively harder.

PURPOSE OF THE GAME

You are a mighty pirate who has amassed a great deal of treasure hidden away in your mountain fortress. The Ninjas, your mortal enemy, have grown extremely jealous of your wealth and wish to steal it from you. You job is to defend your treasure hordes at all cost. The enemy may seem relentless, but I promise you if you hold out long enough they will give up and go home.

INFLUENCES TO OUR GAME DESIGN

The design and implementation of our game is heavily influenced by two different genres of video games. The relatively new Tower Defense genre, and the perennially strong Platformer.

TOWER DEFENSE

Tower Defense games are something of a recent fad in the gaming world, and have become immensely popular in the casual gaming market. Numerous tower defense games exist in various different formats, including web-based flash applications, smart phone apps, and perhaps the most notable example is the multi-platform behemoth Plants vs. Zombies. The premise to a tower defense game is deceptively simple, each game takes place on a map, which has a set (or more) starting point(s) for enemies to spawn, and some sort of target or goal the enemies are attempting to reach. The goal of the player is to let as few enemies as possible reach their goal. The way towers are implemented in these scenarios varies from game to game, some games give you a limited number of towers, but allow you to place them wherever you wish, while other games have statically placed towers, that the player must then choose to activate. Our particular take on tower defense is in the latter category, there will be statically placed towers that the player must choose between.

TWO DIMENSIONAL PLATFORMER

Platforming games seem to have been around forever, and are characterized by puzzles which involve a character jumping from one platform to another in some fashion. Our game will not be a true platformer, as the player will never have direct control over a character's movement, and there is no jump button. That being said, the level design and automated movement of characters most certainly falls in to the platformer category, and is perhaps most heavily influenced by Donkey Kong, further explained below.

DONKEY KONG

While Donkey Kong is a platformer, it's influence on certain design decisions in our game is so heavy that it felt prudent to give the game special notice. Donkey Kongis an iconic arcade game released by Nintendo in 1981, where the player spawns at the bottom of a level and attempts to make their way to the top by climbing ladders and jumping from platform to platform. Our level design and gameplay mechanics map very heavily to this particular title, which one notable exception. In our game, the player controls characters that start at the top of the map, and attempts to prevent enemies from making their way up the platforms. For this reason, we've dubbed this core mechanic in our game "Inverse Donkey Kong."

TECHNICAL CONSTRAINT SATISFACTION

Our design satisfies all the technical constraints in the following ways

- **Hierarchy and Transformation**
  Pirate is a Humanoid Creature, Ninja is a Humanoid Creature. Both creatures will be built using basic primitive shapes that will follow specific transformations to simulate a humanoid body.
- **Primitive Attributes**
  Platforms, Towers, and Treasure Piles will be built using primitive shapes with textures mapped to them.
- **Complexity**
  In addition to the basic humanoid hierarchies of Pirates and Ninjas, simple scene node primitives simulating cannon balls will be flying around the screen and colliding with other objects.
- **Pre-defined animation**
  Pirates will travel along pre-defined paths from tower to tower, as directed by the player. Ninjas will spawn and then autonomously follow a pre-defined path to reach their objective. Cannon balls will be fired from towers within a firing arc which the player has control over. A death animation will be featured when a Ninja is struck by a cannon ball.
- **World Coordinate Window**
  Main View is a zoomed view of the world. It can be manipulated by panning the main view, selecting a new center on the minimap, or choosing a new active tower which will center the view there.
- **Multiple Views**
  We have implemented both a main view and a minimap view.
- **Semantic of Application**
  The game has a clearly defined purpose, outlined previously in this proposal.
- **Audio Support**
  Sound effects will play upon certain object collisions, beginning of a new game stage, or other notable events.
- **Suspend/Resume/Reset**
  All objects will implement functionality to pause and resume their state. A reset button will restore all objects to their initialized state.
- **User Friendliness**
  All User Interface features are clearly and intelligently named and described.

GRAPHICAL USER INTERFACE

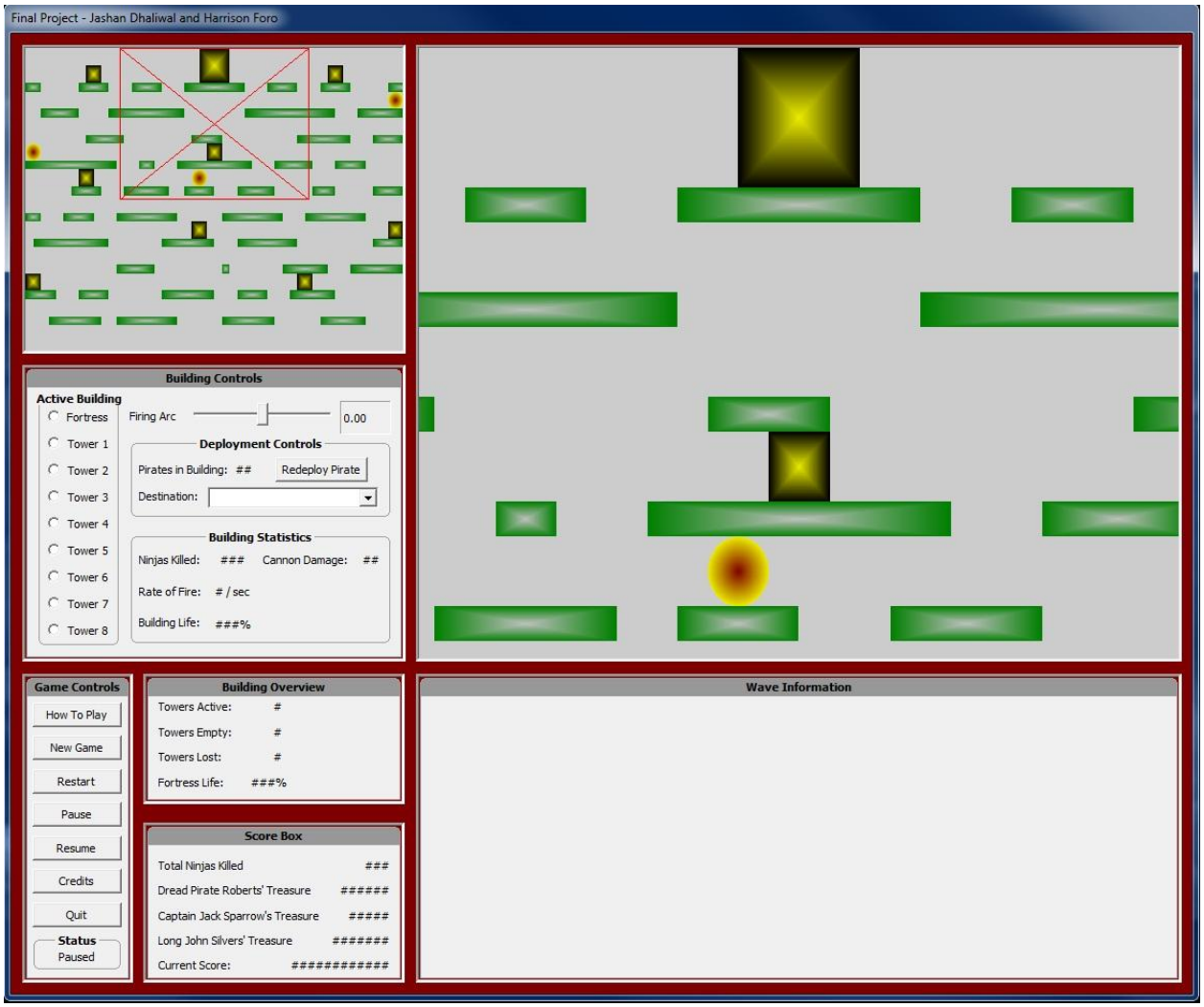

FIGURE 1 PROTOTYPE UI SCREENSHOT

Our UI can be broken in to three main components: the views, the controls, and the information.

THE VIEWS

Our UI incorporates two separate views. The Main View, which occupies the bulk of the UI in the upper right corner, shows a close up view of what is happening in a small section of the world. This view is mapped to the selection box on the MiniMap, which is located in the upper left corner of the UI.

THE CONTROLS

There are two main control sections on our UI. The Building Controls, and the Game Controls. The Game Controls box enables the user to pause/resume the game, quit the game, begin a new game, or open up another window displaying either instructions on how to play, or the design credits. The building controls box provides the main functionality for playing the actual game. Through these controls, the player will be able to re-deploy their pirates to different towers, and adjust firing arcs to better defend against the oncoming Ninja Horde.

THE INFORMATION

There are three main information sections on our UI, in addition to the graphical representations in the views: the Building Overview, the Score Box, and the Wave Information. The Score Box enables the player to see a dynamic representation of how they are doing in filling their objectives. The Building Overview provides a way of determining how your forces are split up without having to check each tower individually. The Wave Information box will display information about what the current stage of the game is, and how much time is left until the next stage is reached.

VIEW/CONTROLLER PAIR DIAGRAM

FIGURE 2 UI SCREENSHOT WITH VIEW/CONTROLLER PAIRS LABELLED

Our design incorporates three separate View/Controller pairs, with no View only components.

- **View/Controller Pair 1**
  This is our main display view which provides the user with a more detailed view of the selected area. In this view a user may select different towers to control, change a given tower's firing arc, or pan the view around.
- **View/Controller Pair 2**
  This is our minimap, it displays the entire model at all times. Mouse interaction with the minimap will alter the location of the world that the main view is focused on.
- **View/Controller Pair 3**
  This is our main dialogue, with controls to adjust the active tower, redeploy pirates between towers, and display of various information about the game state.

OBJECT DESCRIPTION

PIRATE

BEHAVIOR

Pirates are either garrisoned within towers, or following a pre-set animation path between towers.

INTERACTION

When a pirate reaches a tower, they are garrisoned inside and removed from physical view. If a pirate happens to collide with a ninja while moving between towers, one of them will be removed randomly to simulate death in melee combat.

ANIMATION

Will move between different towers as directed by the player, following pre-defined transformation paths.

NINJA

BEHAVIOR

Ninjas are created with a randomized (from a finite set) path to follow from a spawning point to an objective. They then execute their animations to proceed along this path.

INTERACTION

If a ninja collides with a cannon ball, the ninja plays its death animation and is killed. If a ninja collides with a pirate, one of them will be removed randomly to simulate death in melee combat. Upon reaching a destination, a Ninja will disappear in a puff of smoke, having accomplished its goal.

ANIMATION

Ninjas will move along pre-determined paths in an attempt to reach their destination.

TOWER

BEHAVIOR

Towers are fixed objects that are able to shoot at ninjas if they are garrisoned by a pirate. The number of pirates actively garrisoning a tower determine its rate of fire, and the firing arc is set by the player, within a finite range.

INTERACTION

Towers serve as destinations for pirates, and potentially for ninjas.

CANNON BALL

BEHAVIOR

Cannon balls are the projectiles fired by towers. They are created when a tower fires, and destroyed upon hitting a ninja, or exiting the model's edges.

INTERACTION

When a cannon ball collides with a ninja, the ninja is killed.

ANIMATION

When fired, cannon balls will travel along their chosen trajectory until they strike a ninja or the model's edges.

PLATFORMS

BEHAVIOR

Platforms are fixed position rectangles that make up the map. They serve no purpose except to provide graphical cues as to where ninjas are and are not allowed to go.