

QUINQUE
EQUITAS

Design Document for:

SCION OF THE WOLF

The King is dead. Long live the King!

“If, during a development cycle, you do not look at your teammates and say ‘I have no idea what I am doing,’ you are not pushing yourself or your team hard enough!”™

All work Copyright ©2007 by Quinque Equitatas

Written by Amanda Chaffin

Version # 1.00

Saturday, February 21, 2009

Table of Contents

GAME OVERVIEW	4
PHILOSOPHY	4
<i>Philosophical point #1</i>	4
<i>Philosophical point #2</i>	4
<i>Philosophical point #3</i>	4
COMMON QUESTIONS	4
<i>What is the game?</i>	4
<i>Why create this game?</i>	4
<i>Where does the game take place?</i>	4
<i>What do I control?</i>	4
<i>What is the main focus?</i>	4
<i>What's different?</i>	5
FEATURE SET	6
GENERAL FEATURES	6
EDITOR	6
GAMEPLAY	6
THE GAME WORLD	7
OVERVIEW	7
MAZE SYSTEM	7
FIGHT SYSTEM	7
THE PHYSICAL WORLD	7
<i>Overview</i>	7
<i>Key Locations</i>	7
<i>Travel</i>	7
<i>Scale</i>	7
<i>Items</i>	7
<i>Time</i>	8
RENDERING SYSTEM	9
CAMERA	9
GAME ENGINE	9
<i>Overview</i>	9
<i>Menu System</i>	9
<i>Maze System</i>	9
<i>Fight System</i>	9
<i>Player</i>	9
<i>Collision Detection</i>	9
THE WORLD LAYOUT	10
OVERVIEW	10
MAZE MOCKUP	10
FIGHT MOCKUP	10
GAME CHARACTERS	11
OVERVIEW	11
CREATING A CHARACTER	11
ENEMIES AND MONSTERS	11
USER INTERFACE	12
CONTROLLER CONFIGURATION	12

FIGHT HUD	12
MAZE HUD	12
WEAPONS	13
OVERVIEW	13
MUSICAL SCORES AND SOUND EFFECTS	14
OVERVIEW	14
SOUND EFFECTS	14
MUSICAL SCORE	14
SOUND DESIGN	14
SINGLE-PLAYER GAME	15
OVERVIEW	15
STORY	15
HOURS OF GAMEPLAY	15
VICTORY CONDITIONS	15
CHARACTER RENDERING	16
OVERVIEW	16
WORLD EDITING	17
OVERVIEW	17
MAZE EDITING	17
FIGHT EDITING	17
EXTRA MISCELLANEOUS STUFF	18
OVERVIEW	18
LATER IDEAS	18
“IMAGES APPENDIX”	19

Game Overview

Philosophy

Philosophical point #1

Scion of the Wolf combines the traditional elements of both a maze game and a fighting game, with an additional twist of gathering items to use in the fight system.

Philosophical point #2

The game runs on both the PC and the Xbox360 and is written using XNA and C#. We believe in supporting the grassroots movement to put game development back into the hands of the smaller design teams.

Philosophical point #3

When you create some of these overarching philosophical points about your design, say whatever you want. Also, feel free to change it to “My game design goals” or whatever you like to call it.

Common Questions

What is the game?

Scion of the Wolf is a fighting game where the player must defeat his/her brothers and sisters to successfully win the right to challenge the King for the throne. In the game, the player must find his/her siblings in the maze, picking up items to aid him/her during the fights, and defeat them in hand to hand combat. Once the player has defeated his/her three siblings, the player must defeat the King to be crowned ruler of the realm.

Why create this game?

We are making this game as a part of our coursework for the University of North Carolina, Charlotte’s Intro to Game Design and Development under the tutelage Dr. Michael Youngblood. Also, we like fighter and puzzle games.

Where does the game take place?

The land of many paths, Vai’Datha, is an ancient land, where only the strongest survive. Set in the Iron Age, the land is ruled by many Clans, each with their own King. The Clan of the Wolf, Vyu’Sedrin, owns the grassy plains in the center of the Realm.

What do I control?

The player selects the character, one of the four children of the King, they wish to use in the game. Once the player character is selected, the player controls the movements and actions of that character while the AI controls the rest.

What is the main focus?

The player must defeat his/her siblings in order to win the right to challenge the King for the throne. Once that is successful, the player must fight the King.

What's different?

Instead of traditional fighting games, where the player, typically, picks a character and just beats up the other players, usually in a Karate tournament style, this game has a plot as well as traditional RPG and maze elements built into the gameplay. We are also allowing the players (and the AI) to collect game items, which they can use in the combat system. Finally, we are using Shaders to do special effects on the players when items are used, as well as in other places.

Feature Set

General Features

Two different worlds – a fight world and a maze world (tile based)
2D graphics
Custom sound

Editor

Maps are easily customizable
Fight backgrounds are customizable
Make your own maps and fight backgrounds
Make your own player characters

Gameplay

An overhead scrolling maze for players to traverse.
Items for players to collect and use in the fight world.
Side view 2d fighting system.
Two-tiered rock-paper-scissors system: Low attacks hit high-blocking opponents, mid attacks hit low-blocking opponents, and high attacks tend to beat out mid attacks.

The Game World

Overview

The game world consists of two separate types of games – the maze system and the fight system. The player moves through both worlds, using his/her wits and skill to survive.

Maze System

The maze system of the world is simply an overhead view of a tile based maze built from a text file. Items are located in randomized places in the maze as are the AI. Players can use the maze to their advantage to either encounter or avoid the AI. When a player encounters the AI in the maze, the game changes to the fight system.

Fight System

The fighting system is a side view 2D fight system. Players will be able to use the two-tiered rock-paper-scissors fighting system to defeat their opponents. Players can also use any items found in the maze system to freeze, harm, or otherwise effect the outcome of the fight, depending on the item's properties.

The Physical World

Overview

The maze system is built on a tile based engine with an overhead view. The fight system is simply a side view 2D graphic system where the players must defeat their opponents in two out of three fights.

Key Locations

- In order to defeat the siblings, a player must encounter them in the maze. That location is saved when the players go back from the fight system.
- The throne room – once the siblings are all defeated, the player will then be able to access the throne room and defeat the King.

Travel

In both worlds, players walk from one location to another.

Scale

In the maze, the scale of the player is based on the tile width and height. In the fight system, the player scale is the texture height and width which is 100 by 250.

Items

- Freeze – stops the opposing player for five seconds
- Cannonball – throws a cannonball at the opposing player
- Healing Draught – restore health to full
- Slow – cuts the opposing player's movement rate in half, lasts for ten seconds
- Shield – reduces damage, lasts for ten seconds

Time

Time, based on GameTime, is used in both the maze and the fight system. In the maze system, we track how long it takes the player once inside the maze. That number is subtracted from the overall score. In the fight system, we have a 90 second time limit in which the player must defeat the enemy or he/she dies once the time expires.

Rendering System

Since we are using XNA, we will be using the rendering system built into XNA.

Camera

We are using two different camera angles in the game – overhead (maze) and side (fight) view.

Game Engine

Overview

The game engine has three distinct parts – the menu system, the maze system, and the fight system. We control which part we are in by using `GameRules.GameType` in order to switch between the three parts. `Player` is an abstract class that both `AI` and `Human` extend.

Menu System

The screen manager is a component which manages one or more `GameScreen` instances. It maintains a stack of screens, calls their `Update` and `Draw` methods at the appropriate times, and automatically routes input to the topmost active screen.

Maze System

The maze system handles the loading and drawing of the map the players traverse. It also maintains a player list, an item list, and a walkable list. Once a player dies or something is used, it is removed from the respective list. Maze also handles its own `Update` and `Draw` methods.

Fight System

The fight system is a component which manages the actual fights between the players. It handles the HUD for the player, draws the world the players fight in, handles the player collisions and physics, the game state, the game items used, and handles tracking the time of the game as well as player deaths.

Player

The player is an abstract class that both the `AI` and the `Human` use for initialization, update, draw, and various other methods such as `ChangeHealth`, `Bump`, and `RollBackPosition`. We will allow both the `AI` and `Human` to move about the world using the input state manager. Animations are also controlled through the player class as well as the textures, name, status, speed, velocity, scale, and item inventory.

Collision Detection

Using the built in `BoundingBox` in XNA, we will use physics to determine collisions and responses. For the maze and fight system, the bounding box will be the size of the player character image. Also, in the fight system, we will use a weapon bounding box, smaller than the player image, to simulate fists and feet.

The World Layout

Overview

The maze is one maze, tile based, that the players walk around. The fight system worlds are 2D textures set as background images that the players fight on. The fight system will use four different textures that are used randomly in the fights.

Maze Mockup

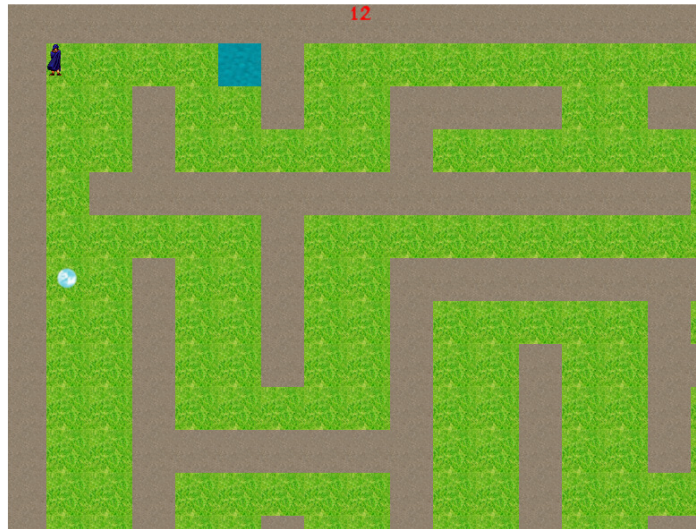


Figure 1 – Mockup of the Maze

Fight Mockup



Figure 2 – Mockup of the Fight, Including HUD

Game Characters

Overview

Players will have the chance to pick between one of the four contenders for the throne. Each character will have personalized attributes and skills that the players can use to defeat the other players, so choose carefully.

Creating a Character

Once in the menu system, players need to go to the player select screen and pick which player they want to use in the game. The three players that they do not select become the AI for the game.

Player Select Screen Mockup

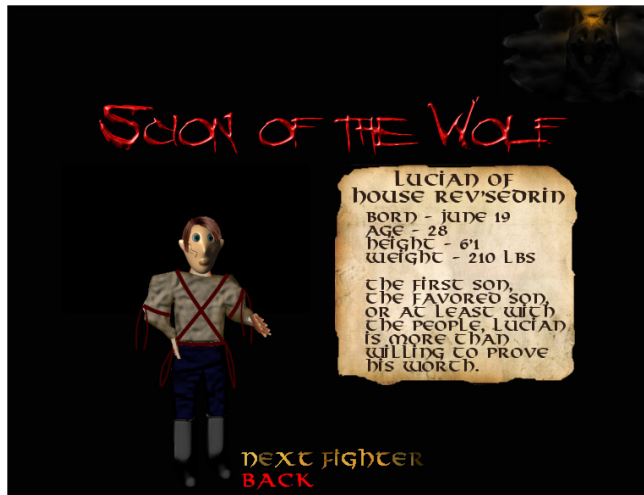


Figure 3 – Player Select Screen Mockup

Enemies and Monsters

The three siblings left after the player selection have the standard AI and their own capabilities. The King, the enemy fought at the end of the game, has the standard AI, his own capabilities, and a few surprises for the player (better items, more health, moves faster).

User Interface

Controller Configuration

The basic controls (seen in Figure 1 below) have the options for players to kick, vertical strike, horizontal strike, and guard. Using the movement to control the high and low attacks, players press the appropriate buttons to launch attacks on their opponents.

Controller Configuration



Figure 4 - Shows the Controller Configuration

Fight HUD

The HUD (see figure 2 below) during battle will show the two player's in battle, their icons, their health, the Human's items, and the time the fights have taken. There is a scoring bonus for ending fights quickly as well as not taking damage during a fight that can eventually be cashed in for fighting upgrades.

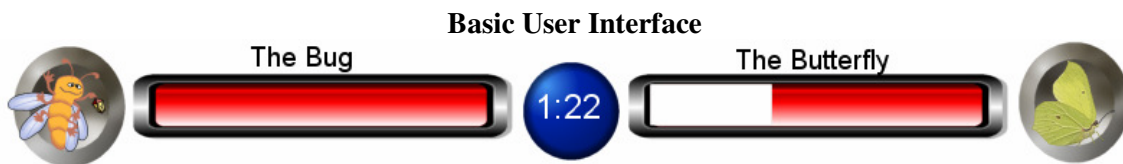


Figure 5- Basic HUD

Maze HUD

The only HUD the maze will have will be the time in the center, the same as the fight HUD. Items will be on the maze itself, until the player picks them up.

Weapons

Overview

The only weapons in the game, with the exceptions of fists and feet, are the game items.

- Freeze – stops the opposing player for five seconds
- Cannonball – throws a cannonball at the opposing player
- Healing Draught – restore health to full
- Slow – cuts the opposing player's movement rate in half, lasts for ten seconds
- Shield – reduces damage, lasts for ten seconds

Musical Scores and Sound Effects

Overview

There are two basic types of sounds in the game – the musical score and the sound effects. Since XNA provides an audio manager (XACT2), we will be using that to put the sounds into the game.

Sound Effects

We are using sound effects to denote changes in the menu system, items selected, decreased health, starting a fight, winning a fight, using an item, as a warning when the time/health is running out, and when the player wins the game. These sounds are going to be simple wav files that we create with a keyboard or a microphone. All object collisions should have appropriate concussion noises and all effects should be at a sublime level, so that repeated play does not become annoying to the players.

Musical Score

Both the maze and the fight system will have their own theme songs written for the piano. The maze theme will be slightly lighter and a little slower than the fight theme.

Sound Design

We will use a "circle of fifths" pattern to transition from one thematic key to another. The use of major, harmonic minor, and melodic minor should reflect real world events.

Pure Scales

Major – Victory; Doing well; Happy

Melodic – Sad; Thoughtful; Deliberative; Driven

Harmonic – Danger; Action; Wry; Scary

Mixed Scales

Major/Melodic – Airy; Interesting; Watery; Space odyssey;

Major/Harmonic – Important/Ending

Melodic/Harmonic – Suspense; Conflict; Climax

Single-Player Game

Overview

In the single player mode, the player will walk around the maze, locating items and the AI. Once the player has found the AI, the game will transition to the fight system where the player will have to defeat the AI two rounds out of three. At the end of the fight, the character will go back to the map world, to collect more items and to find new AI to fight. After all three AI have been defeated, the player will go to the Throne room where he/she will fight the King in a two out of three round battle.

Story

In the great clan, Vyu'Sedrin, there is only the game that matters; the game of thrones. The best player, the King, is aging and needs to name the Heir to the Throne. Following the grand tradition, the King sends his offspring to the furthest corners of the realm with the simplest of instructions – return and claim the throne.

Beat the other players to the throne city and win the chance to rule, if you can survive the journey. Players move around the world in an effort to knock the other players out of the game in battle. Defeat the other players and win the right to be named Heir.

Hours of Gameplay

The game will last anywhere from five to thirty minutes, depending on how the players choose to play the game.

Victory Conditions

Defeat all three of your siblings and then defeat the King. Each fight is two falls out of three.

Character Rendering

Overview

Characters are going to be hand drawn and scanned into Paint Shop Pro (PSP). Once we have the sprite sheets into PSP, we can do any touch ups on the individual sprites that needs to be done and also figure out what the specific bounding range for the sprites for use in the animation system.

World Editing

Overview

The world editor will consist of a simple text editor (for the mazes) and a graphics editor (for the fight system).

Maze Editing

If a user wishes to add a new maze, all he/she has to do is create a new text file with the correct values. We are using a multi level map, where we set the array to hold the terrain layer, the isWalkable layer, the transition layer, and the object layer. With the values set and defined in the game code already, anyone can create a new text file and make a new maze.

Fight Editing

If a user wishes to add new backgrounds to the game, he/she will need to have the source code of the game. All they need to do at that point, however, is create a new background at 1280 by 960 pixels, add it to the game, and recompile the code.

Extra Miscellaneous Stuff

Overview

This is my ‘if I had more time, more people, and money to do what I want to do’ section.

Later Ideas

At some point, I want to make this game multiplayer, or at least have an option for the players to just fight one another. This is relatively easy to do, however, it does put the game out of scope for the current project and the pitch for the game.

Later, I would like to make this a networked game as well as port it over to 3D graphics. Eventually, I plan on pulling it out of XNA entirely and writing it solely in C++ and OpenGL.

“Images Appendix”

Figure 1 – Mockup of the Maze

Figure 2 – Mockup of the Fight, Including HUD

Figure 3 – Player Select Screen Mockup

Figure 4 - Shows the Controller Configuration

Figure 5- Basic HUD