Quest for the Crown

# Game Definitions

Version 1.0 – 25/05/2013

Author: Diogo Muller

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Comments** |
| V 0.1 | 25/05/2013 | Initial Version |
| V 0.2 | 25/05/2013 | Added data about the first iteration. |
| V 0.3 | 27/05/2013 | Updated with first iteration postmortem data and new data for the second iteration. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

## About the Game

Quest for the Crown will be a adventure/exploration game, based on the oldest Zelda games, with some random elements. The whole game will be made using C/C++, in a console environment.

## Basic Story

In this game, you are the prince of a distant kingdom. An evil wizard stole your crown, and if you don’t recover it soon, he’ll claim the kingdom as his own. To thwart the wizard’s evil plan, you will explore dungeons, searching for ancient artifacts that can counter-attack the evil wizard powers.

## Feature Descriptions

The game will be implemented in a cascading way, where the next step will only be implemented after the last step is fully implemented, tested, and deemed stable.

In the most basic scenario, the prince will have one weapon, his sword, and has to kill a few enemies on the screen.

On a second version, the player has to transverse a few sceneries, killing new enemies, until he reaches the evil wizard, a boss enemy.

After this basic feature is done, the next part is creating a dungeon, with a new weapon (bow and arrow), one life power up and pickup, new enemies, puzzles and a new boss character. The player must get an item after the dungeon, and only after that he can battle the evil wizard.

The next step is doing another extra dungeon, with a new weapon (bomb), one life power up, new enemies, puzzles and one new boss and artifact. Only after both dungeons the player will get the opportunity to battle the evil wizard.

After both dungeons are done, the next step is creating random caves/events. Those will show up randomly on pre-determined places on the map. The caves will be procedurally created, and will contain Money, Arrows, Bombs and extra health, if the player isn’t with its maximum possible health.

The last step is creating cut scenes to better tell the story, and a town, where the player can spend his money buying supplies and interact with townspeople, which can give the player interesting tips.

# Game Features

The game features are listed here, ordered by priority. Smaller priority values mean more important features.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Priority** |
| Player movement | The player can move on the screen. | 1 |
| Player attack | The player can attack enemies. | 1 |
| Basic enemy | Basic enemies to be attacked. | 1 |
| Title screen | Game title screen with game name and author. | 1 |
| Game Over Screen | Game over screen. | 1 |
| Basic GUI | Shows player name and health. | 1 |
| Location change | Player movement to other maps. | 2 |
| Different enemies | Enemies with different forms/ways of attacking the player. | 2 |
| Evil wizard | Boss character, evil wizard | 2 |
| Victory screen | A victory screen. | 2 |
| First Dungeon | Basic dungeon, with at least a few rooms and puzzles. | 3 |
| Weapon: Bow | New weapon, bow. It will launch projectiles that run the screen until they hit a wall or an enemy. | 3 |
| Pick Up: Arrows | Arrow pick up and GUI element. The player cannot use the bow if he has no arrows. | 3 |
| Pick Up: Life | Will allow the player to restore one life unity. | 3 |
| Power Up: Life | Will add one unity to the player’s life. | 3 |
| First Boss | Boss character, easier to defeat using the bow. | 3 |
| Long distance enemies. | Enemies that can attack from a long distance. | 3 |
| Second Dungeon | Bigger dungeon, with more puzzles and keys. | 4 |
| Pick Up: Keys | Keys to open closed doors. | 4 |
| Stronger enemies | Enemies that need more than one attack to be killed. | 4 |
| Weapon: Bombs | Bombs will kill every enemy on its blast radius, and can open hidden rooms. | 4 |
| Second Boss | Easier to kill using bombs. | 4 |
| Procedurally generated Caves | Will have enemies and at least one treasure. | 5 |
| Random Events | Can give the player more items, or power ups. | 5 |
| Power up: Extra arrows | Allows the player to carry more arrows. | 5 |
| Power up: Extra bombs | Allows the player to carry more bombs. | 5 |
| Pick up: Money | Money, to be used as points or to buy items. | 5 |
| Store | Store, can be used to buy Power ups and refills, use money. | 5 |
| Cut scenes | Tell the player the game story. | 6 |
| NPCs | Can interact with the player. | 6 |
| Towns | Where the player can buy things and talk to NPCs. | 6 |
| Rebalancing | Game rebalancing, testing the enemies new abilities and where they are placed. | 7 |
| New features | Study new features. | 7 |

# Characters

The game will initially have the main character and one basic type of enemy, with more types being added on the next iterations.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Appearance | Description | Alignment |
| The Prince | @ | You, the main character. | Good |
| Slimes | o | Most basic, weak, puny enemies. | Evil |

# First Iteration

## Time and Activities

Every objective was completed on the first iteration. The biggest part of the iteration was implementation, with documentation also taking a good part of the time. The time breakdown is as follows:

|  |  |
| --- | --- |
| **Activity Type** | **Time** |
| Basic Setup | 0:29:00 |
| Documentation | 1:46:00 |
| Study | 0:21:00 |
| Implementation | 8:03:00 |
| Design | 0:32:00 |

The activities of this iteration where:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Status** |
| Player movement | The player can move on the screen. | Done |
| Player attack | The player can attack enemies. | Done |
| Basic enemy | Basic enemies to be attacked. | Done |
| Title screen | Game title screen with game name and author. | Done |
| Game Over Screen | Game over screen. | Done |
| Basic GUI | Shows player name and health. | Done |

And a more detailed activity breakdown, with times:

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Activity** | **Observation** | **Delta Time** |
| 25/05/2013 | Basic Setup | Project creation, folder structure, library importing. | 0:07:00 |
| 25/05/2013 | Documentation | Game basic definitions. | 0:56:00 |
| 25/05/2013 | Design | Game design decisions and basic design documentation. | 0:32:00 |
| 25/05/2013 | Basic Setup | Git Repository setup and initial commit. | 0:13:00 |
| 25/05/2013 | Study | Console lib study. | 0:12:00 |
| 25/05/2013 | Implementation | First iteration. (Main loop created) | 1:05:00 |
| 25/05/2013 | Basic Setup | Set primitive bug reporting system. | 0:03:00 |
| 26/05/2013 | Implementation | First iteration. (GameObject, Enemy and Player classes, simplified game loop). | 1:37:00 |
| 26/05/2013 | Study | Keyboard input study. | 0:09:00 |
| 26/05/2013 | Implementation | First Iteration. Player movement and positioning control. | 1:08:00 |
| 26/05/2013 | Basic Setup | Git ignore file. | 0:06:00 |
| 26/05/2013 | Implementation | First Iteration. (GameManager) | 1:38:00 |
| 26/05/2013 | Implementation | First Iteration. (Enemy and Player hit, attack, and Health. Game Over condition added.) | 2:13:00 |
| 26/05/2013 | Implementation | First Iteration. (Title Screen and Game Over Screen. First Iteration end.) | 0:22:00 |
| 27/05/2013 | Documentation | Game design document update. (First Iteration detailing) | 0:50:00 |

## Resulting Game

The end result was a simple playable game with title, game and game over screens. As the game starts, the first screen that greets the player is the following title screen:



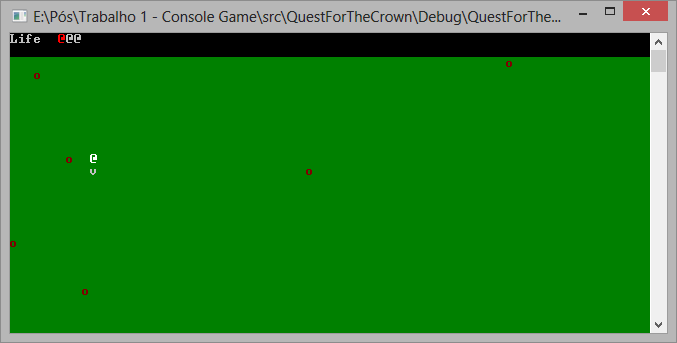
Title screen

After pressing the SPACE key, the game begins, and the player starts the game on a screen with a few enemies. After the game starts, and every time the player is hit, he glows yellow and gets invulnerable for a few milliseconds.



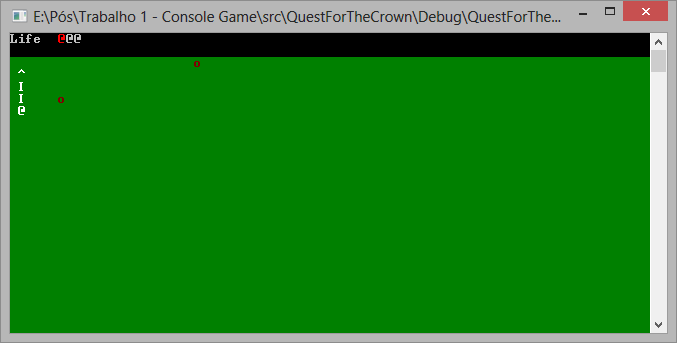
Basic Gameplay – Player Invulnerable State

After those few milliseconds, the player regains its normal color, and can be hit. If the player loses 3 points of life, he dies.



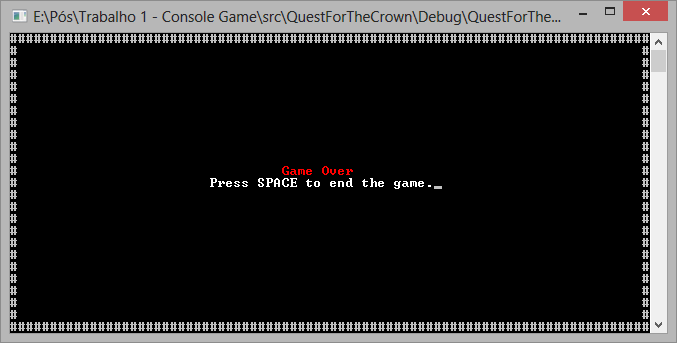
Basic Gameplay – Player Normal State. Notice the player was already hit twice in this case.

The objective of the game is to kill every enemy, and to do that, the player character can use a basic attack with a range of 3 units. If the enemy is hit by any of the three sword parts, it dies.



Basic Gameplay – Attack Animation

If the player loses all of its health or kills all the enemies, the game state changes to Game Over, and the game over screen is presented to the player, where he can hit the space bar to finish the game.



Game over screen.

## Reflections and Lessons Learned

On this first iteration, most of the time was spent with implementation, as expected. The whole iteration took about double the time I thought it would at first, and also some of the Basic Setup items and Documentation took more time than it was first expected, but the final results where far better than what was planned on the start.

A very good lesson that can be learned from this iteration is that I shouldn’t be afraid of redoing big chunks of code, and that sometimes the best code is the one that works well, and you can understand it well, even if it’s not the most efficient or the most elegant solution you could think of.