**Game Design Document**

Version 1.0



**Armalia**

*“When war dies, revenge is born.”*

**Persei Games**

Anderson, Justin

Calvillo, Anthony

DeSilva, Nilmini

Tan, Josh

**Group Project Assignment 2**

**Section 1  
March 1, 2013**

Table of Contents

[Design Document 2](#_Toc349905653)

[Design Guidelines 2](#_Toc349905654)

[Game Design Definitions 2](#_Toc349905655)

[Game Matrix 2](#_Toc349905656)

[Game Flow Chart 3](#_Toc349905657)

[Player Elements 4](#_Toc349905658)

[Player Definition 4](#_Toc349905659)

[Player Properties 6](#_Toc349905660)

[Player Rewards (Power-ups & Pick-ups) 8](#_Toc349905661)

[User Interface (UI) 9](#_Toc349905662)

[Heads-Up Display (HUD) 9](#_Toc349905663)

[Player View 10](#_Toc349905664)

[Antagonistic Elements 11](#_Toc349905665)

[Antagonistic Definitions 11](#_Toc349905666)

[Antagonistic Properties 12](#_Toc349905667)

[Antagonistic List 12](#_Toc349905668)

[Artificial Intelligence (AI) 13](#_Toc349905669)

[Global Game Elements 13](#_Toc349905670)

[The Story 14](#_Toc349905671)

[The Story Copy 14](#_Toc349905672)

[Concept Art 14](#_Toc349905673)

[Level Design 14](#_Toc349905674)

[Level Copy 14](#_Toc349905675)

[Audio & Sound F/X 14](#_Toc349905676)

[Game Architecture 15](#_Toc349905677)

[Game Architecture Overview 18](#_Toc349905678)

[Architecture Copy 18](#_Toc349905679)

[How to Play Copy 18](#_Toc349905680)

[Technical Document 18](#_Toc349905681)

[System Requirements 18](#_Toc349905682)

[Visual Content 18](#_Toc349905683)

[Audio Content 19](#_Toc349905684)

[Programming Content 19](#_Toc349905685)

[General 19](#_Toc349905686)

[Player Elements 19](#_Toc349905687)

[Antagonist Elements 19](#_Toc349905688)

[Code Structure 19](#_Toc349905689)

[Concerns and Alternatives 20](#_Toc349905690)

[Resources 20](#_Toc349905691)

# Table of Figures

[Figure 1 Game Flow Chart 5](#_Toc349905751)

[Figure 2: hud display for level exploration and battle modes 11](#_Toc349905752)

[Figure 3: player view within context of map level 12](#_Toc349905753)

[Figure 4: Game Screen Flow Chart 19](#_Toc349905754)

[Figure 5: UMl diagram 22](#_Toc349905755)

[Figure 6: Uml Diagram 23](#_Toc349905756)

# Design Document

## Design Guidelines

The overall design guideline is to always make design decisions that result in an enjoyable game. Given the time constraints for this game, a recurring theme will be that of simplistic. The primary focus should be on fun game play and an engaging story, not elaborate visuals or complex environments.

## Game Design Definitions

The main purpose of Armalia is for the user to kill the Main Boss (Zane) saving the world of Armalia. Achieving this goal causes the player to win the game. The player loses if he is defeated by any enemy in the course of his/her play through. The player will generally need to acquire a special item to give him access to a new area of the map. Players will be able to access items through the menu to use/equip, and also be able to attack enemies using their spells, which will they will acquire over time as they gain each level. The main focus of the game will be to travel through the various levels and defeating the dungeon bosses for each level. This will lead to the final boss fight (Zane, as mentioned previously).

## Game Matrix

The properties for both player and antagonistic elements are listed in Table 1.

Table : game elements and properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object** | **Properties** |  |  |  |
| *Unit type* | *Hit Points* | *Damage* | *Attack speed* | *Movement* |
| *Basic Enemies* |  |  |  |  |
| Templar (Level I) | 100 | 10 | Slow | Slow |
| Knights (Level I) | 90 | 10 | Slow | Slow |
| Horseman  (Level I) | 85 | 10 | Fast | Fast |
| Assassin  (Level I) | 45 | 8 | Fast | Fast |
| Archer (Level I) | 40 | 5 | Medium | Medium |
| Templar (Level II) | 200 | 20 | Slow | Slow |
| Knights (Level II) | 180 | 20 | Slow | Slow |
| Horseman  (Level II) | 170 | 20 | Fast | Fast |
| Assassin  (Level II) | 90 | 16 | Fast | Fast |
| Archer (Level II) | 80 | 10 | Medium | Medium |
| Templar (Level III) | 300 | 30 | Slow | Slow |
| Knights (Level III) | 270 | 30 | Slow | Slow |
| Horseman  (Level III) | 255 | 30 | Fast | Fast |
| Assassin  (Level III) | 135 | 24 | Fast | Fast |
| Archer (Level III) | 120 | 15 | Medium | Medium |
| Templar (Level IV) | 400 | 40 | Slow | Slow |
| Knights (Level IV) | 360 | 40 | Slow | Slow |
| Horseman  (Level IV) | 340 | 40 | Fast | Fast |
| Assassin  (Level IV) | 180 | 32 | Fast | Fast |
| Archer (Level IV) | 160 | 20 | Medium | Medium |
| *Special Enemies* |  |  |  |  |
| Zane | 3500 | 120 | Fast | Fast |

## Game Flow Chart

The interaction of players, levels, and the components contained within levels is illustrated in Figure 1.

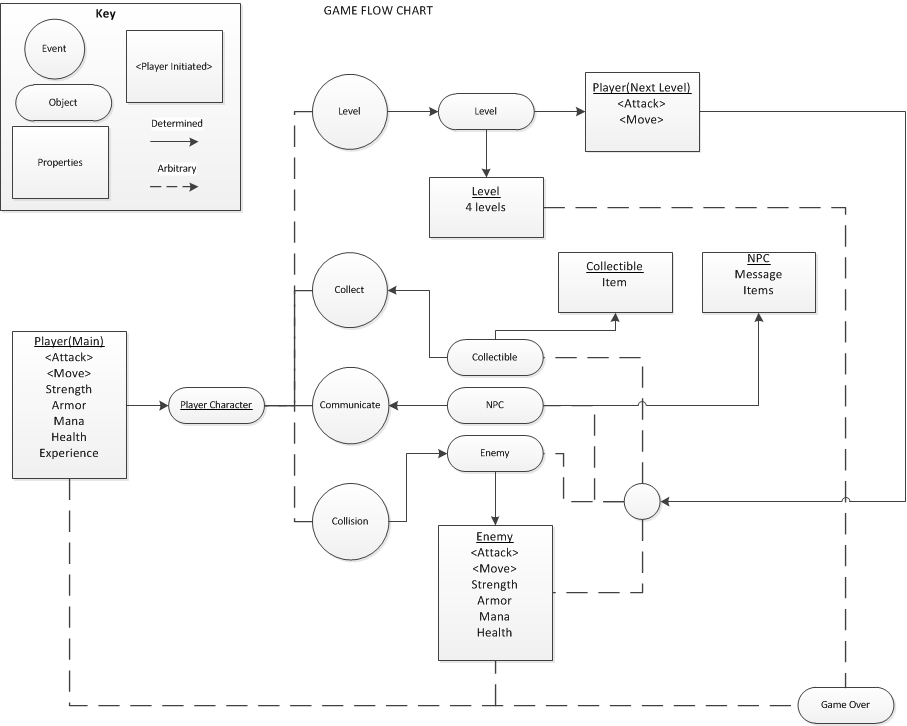


Figure Game Flow Chart

From this chart, we see that the major events between players and enemies involve collecting, communicating, and colliding.

The battle sequence employed corresponds to the usual turn-based battle approach. The sequence can be described as follows:

while Player is alive AND Enemy is alive:

1) Enemy attacks | Player waits

2) Player prompted to select attack

3) Player attacks | Enemy waits

The choice of which side attacks first will be determined by the side that initiated the battle.

## Player Elements

### Player Definition

***Actions***

Players will have separate sets of available actions, depending on the whether they are exploring the level or engaged in battle with an enemy.

*Global:*

Players will be able to pause the game at any time using the escape key. In the pause screen, players will be presented with a menu that allows the game to be exited. In addition, a previous save point can be loaded from this screen.

*Exploration:*

Movement - Players can move their character left, right, up or down. The ability to move diagonal will not be available, given the tile-based nature of the game map. In addition, movement will restricted to constant velocity; running through a map will not be possible. This will simplify the game mechanics and help to avoid characters running in order to avoid enemy battles.

Character Customization - Players will have the ability to customize their characters weapons and armor via on on-screen menu. Menu items can be selected using the mouse as input, though keyboard shortcuts will also be made available for the more common menu items.

Item Usage and Management - Items such as potions or antidotes can be used by selected the appropriate buttons from the on-screen menu. Items can also be discarded using the menu, in order to free up available slots.

NPC Interaction - Players can interact with NPCs by pressing a specific key when their characters are near and facing the NPC. Another way of initiating this interaction will be to click on the NPC with the mouse, provided the character is sufficiently near to the NPC. This will display text in a specific location on the game screen, corresponding to predefined game dialogue. In some cases, the player will be able to make some decision in response to the NPC interaction. In these cases, a menu will displayed with options that can be selected using the mouse. NPCs that can be interacted with include certain enemies, allies, and merchants.

Environment Interaction - Players can use either the keyboard or mouse to interact with specific elements of the environments, such as opening up chests (containing coins, items, weapons, or armor) or picking up visible items placed in open chests.

*Battle*

Regular Attacks

Once engaged in battle, the on-screen menu will change to display only those options relevant to game battles. A sub-menu will allow the player to select from a list of available attacks.

Special Attacks

If players have enough mana points, they will be able to execute special attacks on the enemy. Selecting these attacks will replace the game screen with a new one, in which a timing-based challenge will be presented. The timing-based challenge will be reminiscent of the game play style used in the Guitar Hero games. Certain symbols will flow across the screen, top to bottom. Once the symbols cross a specific area near the bottom of the screen, players will need to press the corresponding key. If the key is pressed at the correct time, it will be counted as a hit and the attack power will be increased. When the player completes long successions of successful hits together will result in an extremely powerful attack against the enemy. After the timing-based challenge is finished, the screen will revert back to the normal battle mode.

**Information (Status)**

The player will have access to the following information at all times:

* Chosen character name
* Current character HP, MP, and XP points
* Acquired weapons, armor, and items, as well as which the particular weapons and armor that are
* currently equipped

During battle, the player will be shown all available attacks, including the required MP for attacks (where applicable).

**Default Properties**

The player will begin the game with 100 HP and 100 MP. In addition, the player’s character will start the game with a basic sword as the equipped weapon. Players will be prompted to choose their character’s name at the onset of the game.

**Winning**

Winning the game is accomplished by successively progressing through each of the four levels and defeating the final boss enemy in battle.

**Losing**

Players lose the game whenever their character is defeated in battle. When this occurs, players will have the option to restart the game from the most recent save point.

### Player Properties

Name

The player-chosen name, specified at the beginning of the game.

Hit Points [HP]

The amount of HP is the primary indicator of the main character’s health. The amount of HP that a character has will be visually reflected in the appearance of the on-screen character.

Mana Points [MP]

The amount of MP controls the player’s ability to execute certain attacks. For example, the timing-based special attacks will require a specific amount of MP in order to be executed.

Strength Level

The attack level is a numerical indicator of the strength of a character’s attacks in battle.

Defense Level

The defense level is a numerical indicator of the defense of a character against enemy attacks in battle.

Character Level [LVL]

The level of a character is directly related to the Strength and Defense attributes for that character. In additions, it will also affect the character’s HP and MP capacity (i.e. the maximum possible HP or MP attainable). Reaching certain levels will unlock new attacks that can be used in battle.

Experience Points [XP]

Experience points are acquired by successfully defeating enemies in battle. Certain amounts of XP are required to reach new character levels. The amounts required will increase as the character level increases.

Currency [COIN]

Coins can be acquired in battle, as part of the story, or during level exploration. It can be used to purchase additional items, weapons, or armor from merchants (implemented as NPCs) in the game.

Weapons [WPN]

Various weapons can be acquired in the progression of the game. Characters can have many weapons, but only one weapon can be equipped at any given time. Weapons can categorize into those that are magic-based and those that are not. Magic-based weapons will require MP in order to be used in battle. The rarity of obtaining a particular weapon will correspond to its power.

Armor [ARM]

Players will have the ability to use armor to increase their defense against attack. Multiple pieces of armor can be used simultaneously, provided they apply to different body parts. The types of armor include helmets, body armor, hand armor (e.g. gloves), and leggings. Similar to weapons, armor is distinguished between those that require magic and those that do not.

Items [ITEM]

Items can be classified as consumable or durable. Consumable items are those that are used once and then discarded (e.g. potions). Durable items can be used repeatedly (e.g. a grappling-hook-like hook shot).

Inventory

Items will be stored in and accessed from an inventory. The number of inventory slots will be limited.

Base Difficulty Level

Players will choose a base difficulty level at the onset of the game. This will affect the difficulty of the enemies encountered through the game levels.

Status Effects

A player’s character may be inflicted with certain status effects, resulting from enemy attacks. These status effects will persist after the battle and remain in effect until the player takes some counteractive measure. For example, an enemy may cast a spell that inflicts a curse on the player’s character. A cursed player will continuously lose HP after battle, as time passes. The player will need to use a special item to remove this curse.

### Player Rewards (Power-ups & Pick-ups)

Table 2 shows the different items that can be obtained and used in the game, along with their corresponding effects.

Table : Items and their effects

|  |  |
| --- | --- |
| **Item** | **Effect** |
| Potion | Increase HP by a fixed amount, up to a character’s HP capacity. |
| Ether | Increases MP by a fixed amount, up to a character’s MP capacity. |
| Demon Dust | Temporarily increases a character’s strength and defense attributes. Also has the negative side effect of decreasing the maximum hit points for character (i.e. length of health bar). |
| Angel Dust | Restores damage done to a character’s maximum HP through the use of Demon Dust. Very rare. |
| Elixir | Removes curse effect from a character; character will no longer lose health as time passes. |
| Water Shield | Allow passage through otherwise impassible fire barriers. |
| Mana Accelerators | Accelerates the rate at which a character’s MP is regenerated. |
| Magic Scrolls | Temporarily increase damage done by magic-based attacks. |

### User Interface (UI)

Players will control the movement of their character using the keyboard. Actions or menu options will be performed using the mouse. In some cases, keyboard shortcuts will be provided for commonly-accessed menu items, in addition to allowing for selection with the mouse. Table 3lists the mappings between game events and the inputs that trigger them.

Movement

Table : movement

|  |  |
| --- | --- |
| Movement |  |
| Move Left | ‘A’ |
| Move Up | ‘D’ |
| Move Right | ‘W’ |
| Move Down | ‘S’ |
| Environment Interaction |  |
| Initiate NPC Interaction | ‘Space’ or Left Mouse Click on NPC |
| Complex NPC Interaction | Menu System using Mouse |
| Battle |  |
| Select Attacks or Use Items | Menu System using Mouse |
| Timing-Based Special Attack | Uses ‘J’, ‘K’, ‘L’, and ‘;’ (Multiplier Mode\* initiated using ‘Space’) |

\* *Multiplier Mode becomes available if a certain number of notes are successively hit within a single streak. This will multiply the increase in damage for the attack.*

### Heads-Up Display (HUD)

The HUD for the player will be displayed on the right portion of the screen. This section will persist throughout the entire game, although its contents will change dynamically, depending on the game context. The two primary modes of game play are level exploration and battles. The information and menu system shown in the HUD for these two modes are illustrated in Figure 2.



Figure : hud display for level exploration and battle modes

The primary differences between the two modes will be the replacement of the level map and weapon/armor menu in the level exploration mode with the enemy stats and attack menu in the battle mode.

### Player View

They player view will be similar to that used in the classic Legend of Zelda games. Players will view their character in a top-down perspective. The size of the player view will be much smaller than the overall map their character is in. Only a limited area of the map will be in view at any given time. This concept is illustrated in Figure 3.

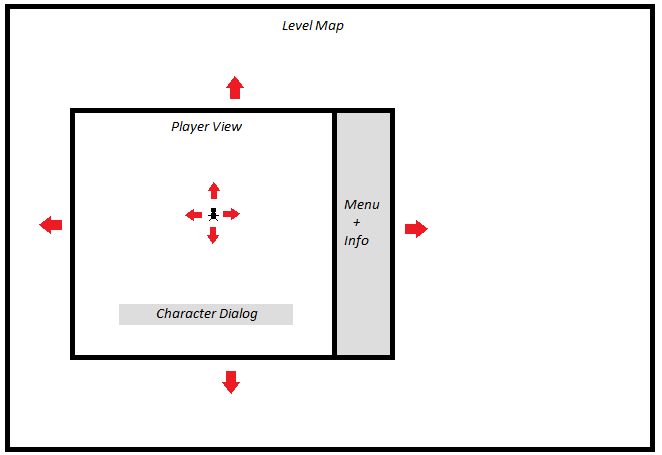


Figure : player view within context of map level

The player’s character will usually be centered within the window viewed by the player. As shown by the red arrows, this window will follow the movements made by the character. An exception to this behavior is when the player is near an edge of the map and moves in such a way that centering the window character would require part of the window to be located outside the limits of the map. In this case, the player view will remain within the bounds of the map and the character would no longer be centered in the map window.

Changes in the visible portion of the game map will be fluid; a single step by the character will move the visible portion of the map by the step size, given that the character is not at the edge of the map already.

## Antagonistic Elements

### Antagonistic Definitions

Antagonist elements can be divided into two main components: enemies and path barriers.

Enemies

Enemies will be spread throughout each game level. They are only hostile towards the player and will move to attack the player’s character if he is spotted. The player is not able to talk to, purchase items from, or sell items to enemies.

Path Barriers

Certain paths in a level will be blocked by magical barriers. In order to pass through these barriers, players will need to use special items that are obtained either in battle or in an interaction with some NPC. For example, a fire barrier will not be able to pass through unless the player’s character has used a Water Shield item on himself.

### Antagonistic Properties

The antagonistic properties for enemies include all those specified for player properties. Items, weapons, armor, and/or currency may be dropped by enemy as a result of defeat. All of the enemies are non-magic units (with the exception of Zane). Each enemy will move in a randomized pattern until detection of a player’s character has been achieved (with the exception of Archers, who will initiate battle from a distance). All enemies have special attacks that are determined by their class.

### Antagonistic List

*Table 1* describes the types of enemies that will be encountered in the game.

Table : enemy descriptions

|  |  |
| --- | --- |
| Templars | Use silencing spells to prevent players from casting spells. Resistant to magic-based attacks. Uses melee attacks to damage player. |
| Knights | Use basic melee attacks. Weak against magic, but resistant to physical damage. |
| Horsemen | Higher rate of movement. Can use a charge attack to inflict high damage, but with a limited success rate for hitting the target. After charging, will engage in melee attacks. |
| Assassins | Attacks are increased in power when executed behind a target. Use poison to damage player over time. |
| Archers | Use concealment to hide from player until they detect the player and battle is initiated. Resistant to range attacks, but are weak against melee attacks. |
| Zane (Final Boss) | Uses Elemental spells i.e. Fire/Ice/Lightning/Earth. Resistant to Magic, and can cast a mana shield to reduce melee damage. |

### Artificial Intelligence (AI)

Enemies can exist in one of four AI states: Normal, Detection, Battle, and End states. These states are described in Table 5*.*

Table : Enemy AI states

|  |  |
| --- | --- |
| *AI State* | *Description* |
| Normal | Enemy will move in a random pattern until it detects the player (except for Archers, who will be concealed in the map and appear when player is detected). |
| Detection | All enemies will detect the player when it enters a radius around the player. The enemy will then proceed to attack the player. |
| Battle | The enemy will attack until either the player or it defeated itself. |
| End | The player will either die, resulting in the game over event or the enemy will die and the player can continue on in the map. |

## Global Game Elements

Boundaries

Players will be restricted to moving to their characters along specific paths within game levels. In most cases, the path will correspond to an actual path or road in the game. In other cases, the path will be implicit. Certain objects in game levels will also serve as a form of player boundaries. Whether or not an object is something that can be moved over will be obvious from the context, e.g. players will not be able to walk over or through large trees.

Neutral Objects

The primary type of neutral objects will be those used for personalizing levels. For example, building and mountains will be found in certain levels, but will have no affect on the player other than to provide visual enhancement of levels.

Camera Views

Only one camera view will be available to players: a top-down, bird’s-eye view. This same view will be used for both level exploration and battles.

Scale of the World

The scale of the world in which Armalia takes place will be vast. However, the player will only experience 4 different sections of this world. Transporters will be used to move the player from level to level, once the end of a level has been reached.

## The Story

### The Story Copy

See attached documents for story line and dialogue.

## Concept Art

A sample of the concept Art created for Armalia can be viewed in the attached game history document.

## Level Design

Levels will be experienced by the player in the following order:

1. Hometown - Starting point of the game and hometown of the main character (Simon)
2. Underground Cave - A level that challenges players by limiting their field of vision. A torch can be used to light up limited parts of the map. In this level, it will be easy to accidentally trigger battles.
3. Forest - A maze-like level in which paths may lead to dead ends, requiring to players to retrace their steps.
4. City of Magic - Final level. Home of Zane, the final boss enemy.

Special items may be recovered to unlock access to a particular level.

### Level Copy

See *attachments* for a copy of the game script (used within levels to reveal the plot).

## Audio & Sound F/X

This game will rely on freely available audio files for sound effects and background music. Looping background music will be continuously played, the type of which depends on the specific level a player is located in.

Simple sound effects will player in battles for attacks and when damage occurs. In addition, an audio clip will be played whenever a special game object is acquired.

## Game Architecture

The following table describes the screens numbered in the flow chart after the table. Table 6 depicts the game flow pattern regarding the game screens.

Table : Game Flow pattern of screens

|  |  |
| --- | --- |
| *Screen No* | *Function* |
| *1* | *Splash screen with logos and about.* |
| *2* | *Title screen that has the start option.* |
| *3* | *Options:*   * *Game - Starts a new game* * *Load Game - Loads an existing game* * *Options - Loads the game options menu* * *Instructions - Loads the game instructions menu* * *Hall of Fame - Loads the high scores* * *About Us - Loads the about us information* * *Quit - Quits the game* |
| *4* | *The player can enter a name and choose the difficulty through the difficulty option. The back option loads the previous screen.* |
| *5* | *The player can load a previous game and the back option loads the previous screen.* |
| *6* | *The player can choose game options. All the options are in the default mode and player can change these if necessary. The key configuration loads the key configuration options page and the back option loads the previous screen.* |
| *7* | *The player can choose different keys to move and do certain tasks.* |
| *8* | *Instructions about the game.* |
| *9* | *A list of high scores.* |
| *10* | *Information about the game developers and development.* |
| *11* | *Different game difficulty levels.* |
| *12* | *The intro sequence animation introducing the game.* |
| *13* | *Game start.* |
| *14* | *Game level I.* |
| *15* | *Game Level II, accomplished through completion of level I.* |
| *16* | *Game Level III, accomplished through completion of level II.* |
| *17* | *Game Level IV, accomplished through completion of level III.* |
| *18* | *The user can save the game.* |
| *19* | *End sequence played after game over with credits. The stop option stops the credits and the menu option reloads the game menu.* |

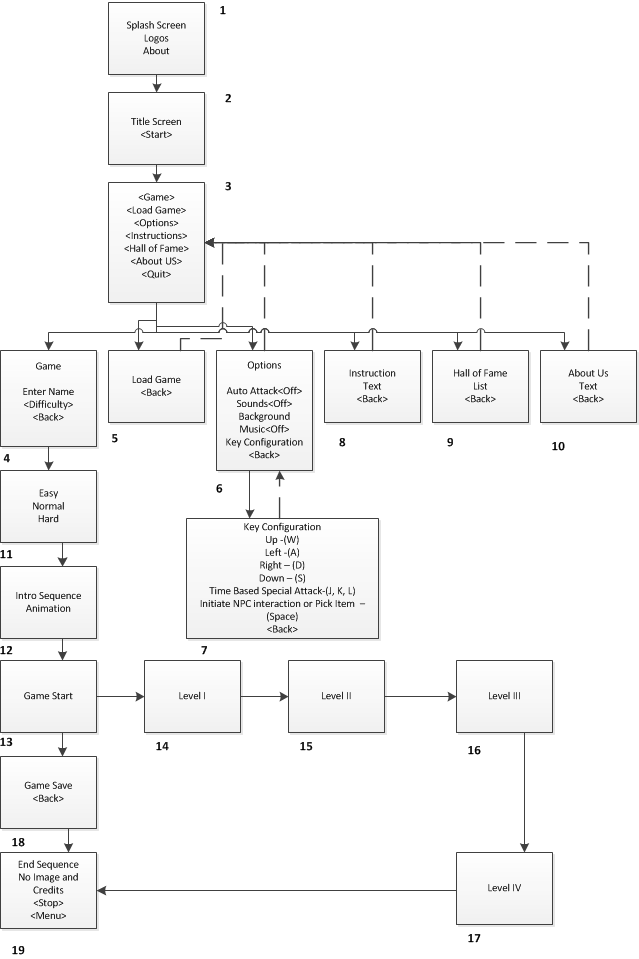


Figure : Game Screen Flow Chart

### Game Architecture Overview

The splash screens or video clip need to be in accordance to game story and style. If cut scenes use video then story boards should be created. This will need to be created with the graphic designer. Menu should be designed with the most important options easily accessible. Be aware how many clicks it takes to accomplish a task. It would be wise to get together with someone from quality control and a programmer. The Game Instructions should be written so that the player understands how to play the game. Mock-ups should be made so that the programmers get the correct layout of the menu. Mention and describe high score screen here as well.

### Architecture Copy

Game is given by the story. See attached screenplay.

### How to Play Copy

This will be done near the game’s completion due to possible changes and modifications through the development process.

# Technical Document

## System Requirements

* Windows XP, 7, or 8
* GPU that supports DirectX 9 and Shader Model 2.0 (necessary for Reach profile)
* At least 512MB Memory
* At least 50MB Hard Disk Storage

## Visual Content

* General
  + File Size Restrictions: none
  + File Format Type: PNG
  + File Quality Type: Good
  + Visual Scale: None
* Player Elements
  + Type of States (Default, Damage, Destroyed, ect.)
    - Moving Left, Right, Down, and Up
    - Player death
    - Player take damage
* Heads Up Display (HUD)
  + Type Icons: TBD
  + States: TBD
  + Font Type: TBD
* Antagonistic Elements
  + Type of States (Default, Damage, Destroyed, ect.)
    - Moving Left, Right, Down, and Up
    - Player death
    - Player take damage
* Global Elements
  + Background/Texture/Tiles: TBD
  + Font Type: TBD

## Audio Content

* General
  + File Size Restrictions: None
  + File Format Type: MP3
  + File Quality Type: Good
* Player Elements
  + Type of Sound f/x
  + Device Vibration: None
* Antagonistic Elements
  + Type of Sound f/x: TBD
  + Device Vibration: None
* Global Elements
  + Ambient Music
* Splash Screens
  + Ambient Music
* Menus
  + Type of Sound f/x: None

## Programming Content

### General

* File types used: C# Files (source code), XML (level development), PNG (graphics), MP3 (sound clips)
* File Sizes: There will be no file size requirements because it won’t be an issue for this game’s design specs.
* Coding Conventions: We will use standardized C# standards. That is, use capital letters for each word (including beginning word). For class structure and methods see attached UML diagram.
* Language Restriction: We will strictly use C#.
* Screen Type: We will use a medium size screen. Exact specifications are yet to be determined.
* Device Restriction: We will be strictly developing for the PC.

### Player Elements

**Event Types:**

* Click event.
  + Allow the player to chose an option in our menu
  + Allow the player to attack an enemy
  + Allow the player to open a chest
* Button Event
  + The player can use the A, W, D, and S to move the character
    - A = Left
    - W = Up
    - S = Down
    - D = Right
* Collision
  + The player collides with an object

### Antagonist Elements

**Event Types:**

* Collision
  + The enemy collides with an object

## Code Structure

See UML class diagram.

## Concerns and Alternatives

There are some decisions left to be made on the code structure. These decisions deal with how to design and operate the menu system for the player. These decisions will be explored once the map and player components are finished (these components are seen as more important).

## Resources

Tiled – 2D Map Editor

XNA Game Studio 4.0 using Visual Studio.

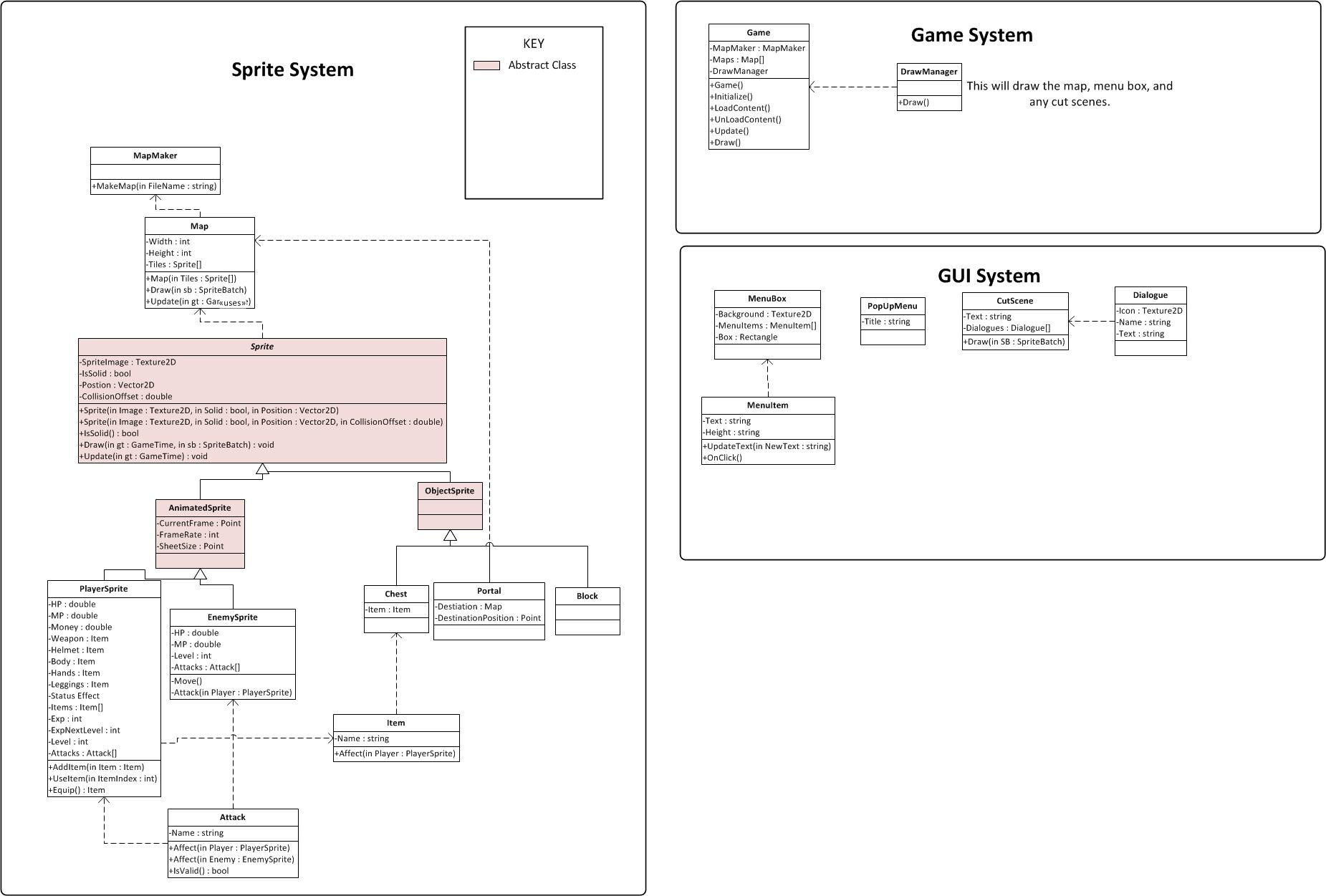


Figure : UMl diagram

## 

## C:\Users\Justin\Documents\School\Csci313\Armalia\armalia\Documents\UMLDiagram.jpg

Figure : Uml Diagram