SYSTEM REQUIREMENTS SPECIFICATION AND ANALYSIS REPORT

Team: NOVART

Members: 1203355 İbrahim Özbay

1250588 Muhammet Zahit Özcan

1250711 Serhat Solak

1203231 Yunus Kocaman

TABLE OF CONTENTS

1. INTRODUCTION…… 3
   1.1 System Purpose…… 3
   1.2 System Scope…… 3
   1.3 Definitions, Acronyms and Abbreviations…… 3
2. GENERAL SYSTEM DESCRIPTION……..……...……... 4
   2.1 System Modes and States……..……...……... 4
      2.1.1 Game Startup ……..……... 4
      2.1.2 Data Driven Mode: ……..……... 4
      2.1.3 Stand By……... 4
      2.1.4 Input……...….. 4
      2.1.5 AI…….. 4
      2.1.6 Sound Effects and House Keeping…… 5
      2.1.7 Rendering…… 5
      2.1.8 Exit…… 5
   2.2 Major System Capabilities…… 5
      2.2.1 Out of the game……..…… 5
      2.2.2 During the Game…… 6
   2.3 Major System Conditions…….…… 7
      2.3.1 Gameplay Related……..…… 7
      2.3.2 Technical Conditions…… 8
   2.4 Major System Constraints…… 8
      2.4.1 Hardware Constraints…… 8
      2.4.2 Gameplay Constraints…… 9
   2.5 Assumptions and Dependencies…… 9
   2.6 Operational Scenarios…… 9
   2.7 Use Case Diagrams…….…… 10
3. SYSTEM CAPABILITIES, CONDITIONS AND CONSTRAINTS….. 14
   3.1 Environmental Conditions In Game…… 14
   3.2 Game Security…… 14
4. USER INTERFACES…… 15

1. INTRODUCTION

1.1 System Purpose

The purpose of our project is to create a 3D single player arcade adventure game, in which the player will spend an enjoyable time period solving puzzles, making decisions, enduring challenges in a realistic environment.

1.2 System Scope

"The Flee from Alcatraz" is a first person shooter game which can be identified generally as a "hunted type" game, however during the course of the game the hero will sometimes be a hunter and sometimes be hunted. It will be a single player game. The application of the system is a window where the rendered scenes of the environment is displayed according to the input given by the user and the state/mode of operation.

The main objective of the game is to help the hero escape from Alcatraz prison by eliminating or sometimes directly running away from guards, reaching some predefined score point regarding the difficulty level, improving abilities, and solving the required puzzles for the level.

1.3 Definitions, Acronyms and Abbreviations

GUI: Graphical User Interface

SyRS: System Requirements Specification

DD: Data Driven: A technique to seperate code and data so that with only changing the external data you can have different outputs in the system.

Hotkey: Combination of keys for doing some predefined action.

Hitpoint: A number given to a character which identifies his health.

Agility: A number given to a character which determines how quick a character is in terms of movement, using weapons etc.

Defence: A number given to a character which determines how resistant a character is.

XP (Experience points): A number collected by a character when he finishes the required puzzles, jumps levels.

2. GENERAL SYSTEM DESCRIPTION

2.1 System Modes and States

2.1.1 Game Startup:

In this mode all the storage needed for the display of the main menu is set. Variables are initialized with the default values, data structures for holding the map and weapon information are set up, necessary memory is allocations are done, and lastly graphics and sound files are loaded by the system.

2.1.2 Data Driven Mode:

In this mode the player first chooses the character, then the difficulty and lastly the level. In this mode all the data needed for the gameplay will be loaded into main memory. Loading a game is a combination of choosing of the map, character and difficulty. Therefore loading and saving are also included in this mode.

2.1.3 Stand By

This is an intermediate mode, to either go into the next mode for input retrieval from player or exit to OS.

2.1.4 Input

The user is prompted for inputs from mouse and keyboard. According to the input of the player either the main loop will suspend and go to the DD mode for main menu display if the user presses escape, save or load buttons or it will stay in main loop for further processing of input.

2.1.5 AI

Based on the last input of the player AI will be applied. After the application of AI, the moves such as bending, stepping, opening of all the objects will be done.

2.1.6 Sound Effects and House Keeping

After AI mode is complete, the required sounds will be produced (i.e. when the player hits the attack button, a gun fire will be heard if he has a weapon.). Also all the changes in the variables and datastructures will be applied, such as when the player moves, the location of the player in the map is changed so the location info is refreshed in the map data structure.

2.1.7 Rendering

Current frame is drawn to the back frame buffer with all the changes produced by the AI mode and then displayed on the screen by changing the front and back frame buffers.

2.1.8 Exit

All the resources allocated will be released and the program will return to OS.

2.2 Major System Capabilities

2.2.1 Out of the game

The game will be capable of processing multiple button presses (Hotkeys) for a more efficient and easier game play. The user will be able to choose his or her hotkeys before entering the main loop of the game.

The user will be able to choose different characters each having different attributes and game AI logic will interpret these attributes and behave accordingly. i.e. if the player has a higher agility the AI will process it such that the player can move faster than the character with less agility.

To make the player familiar with the game play and the buttons used, a simple training mode will be available. In this mode player will learn how to shoot, walk, bend, and will get used to using the map window to find his location with basic instructions given by the game for general moves.

The player will be able to adjust both the ingame sound level and music sound level and also he/she will be able to turn sound on/off.

2.2.2 During the Game

The hotkeys are all held in a data structure so that when the player presses some hotkey the AI will be able to process what it has to do by fetching the required actions from the data structure.

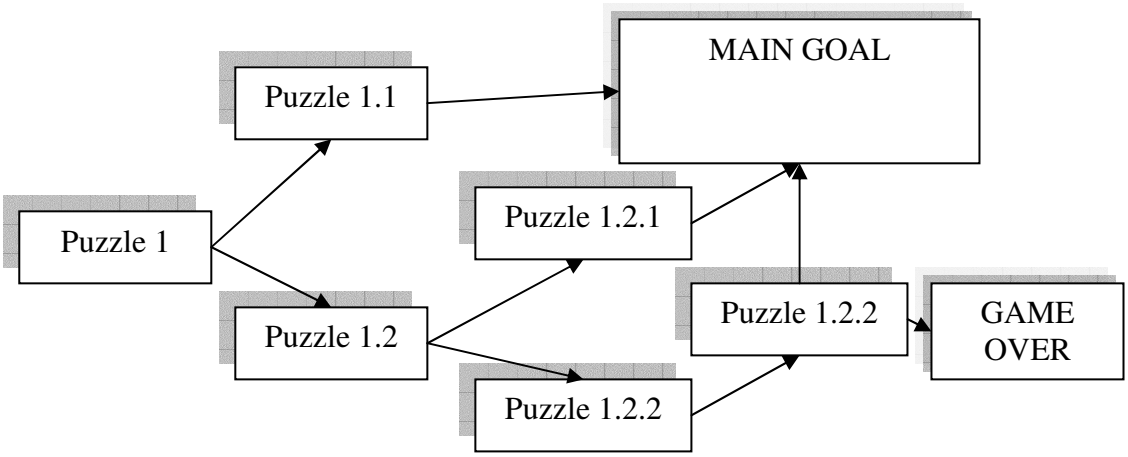


Figure 2.1

Unlike most FPS games there won't be a single path to achieve victory, the main goal. Each path has different features, for example 1 → 1.1 has less puzzles to solve but the player faces more enemies. On the other hand 2 → 2 → 2.2 → 2.2 → 2.2.1 has more puzzles to solve but less enemies to eliminate per puzzle solving stage. Therefore each path has advantages and disadvantages. The best path can change for each hero. The 2nd path seems better for the character with more agility but if you choose the character with more weapon skills the 1st path seems better.

If the player finds difficulties in solving the puzzle there will be hints to help him but the hints won't be offered unless user wants to see them.

The player starts the game without weapons then as the game proceeds, he/she will collect a variety of weapons and bullets from the enemies he/she kills.

The skills of the character will effect how good the character uses each weapon. One character will be very skilled in using the sniper rifle, so from the user point of view to aim will be easier than a character without the sniper skill. As the player passes levels, eliminates enemies he gains XP and he/she can distribute these XP to different attributes.

2.3 Major System Conditions

2.3.1 Gameplay Related

The enemies will be able to communicate with each other. i.e if you are seen by an enemy that enemy will call the nearest enemy for help, by shouting out loud. The other enemies in some predefined area will hear the shout and come to help.

The character can only carry some amount of weapons. The weight of the weapons carried will effect how the character moves, shoots, etc.

The enemy with hitpoints reduced by our character to less than a predefined value will try to escape from the character. During the time they are away (not visible to us), they regain their health gradually and come back to their original stand point on the map.

There will be health packs on the map, which will be used by the player to regain some health. The character will not be able to carry the packs to the next level.

If the player is hit the screen will turn to reddish and back to normal. When the user presses map from main menu or from hotkey, the main loop will suspend and the map will be displayed in a new window overlapping the game display window.

Only the parts of the map that have been visited can be seen, the other parts are displayed black.

There will be a pause button and when it is pressed the game will freeze.

A high score list will be implemented which will hold the player's name, date of the game played, and the score.

2.3.2 Technical Conditions

When the user presses the escape button the main loop will suspend. The game can be played in fullscreen only.

To make our game more realistic we will need realistic models for enemy appearance. We will try to find models from the internet and load them into the game. We will use a program to load them.

For creating sound effects, editing existing sounds, adding effects like echo to them, we will use a sound digitizer called Gold Wave. To create realistic textures for the walls we will need a graphics tool like Photoshop. We won't use any physic engine but we will implement a bit of collision detection and creating gravity. For the background music we will use a music editor.

2.4 Major System Constraints

2.4.1 Hardware Constraints

We expect that the game will require 128 MB ram, and some graphics card supporting OpenGL commands with 64 MB ram, and some processor such as Pentium IV 1GHz.

The game can be played with only mouse and keyboard (no joystick).

2.4.2 Gameplay Constraints

The resolution can not be changed, the game will be played in fullscreen only. There won't be multiple views, the screen will be rendered only from the forehead camera of the character. The user can not determine the order of tackling each puzzle. The sequence is determined beforehand.

There should be a reasonable response time for displaying response to user actions.

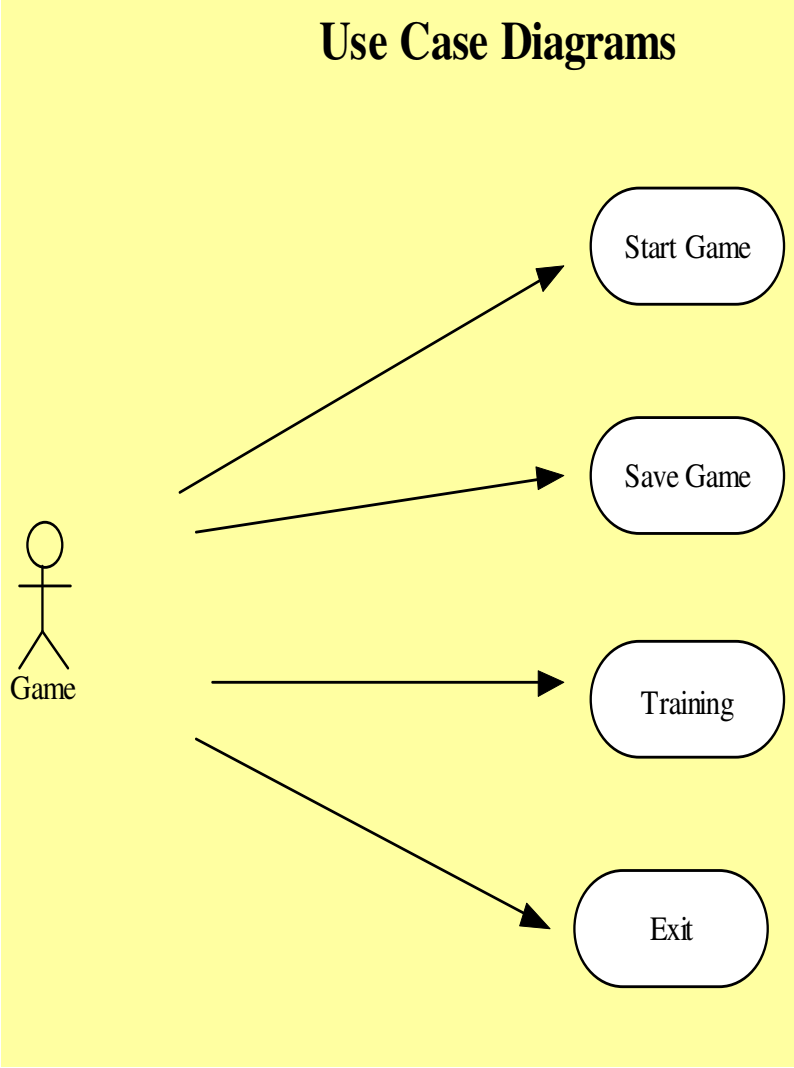
2.5 Assumptions and Dependencies

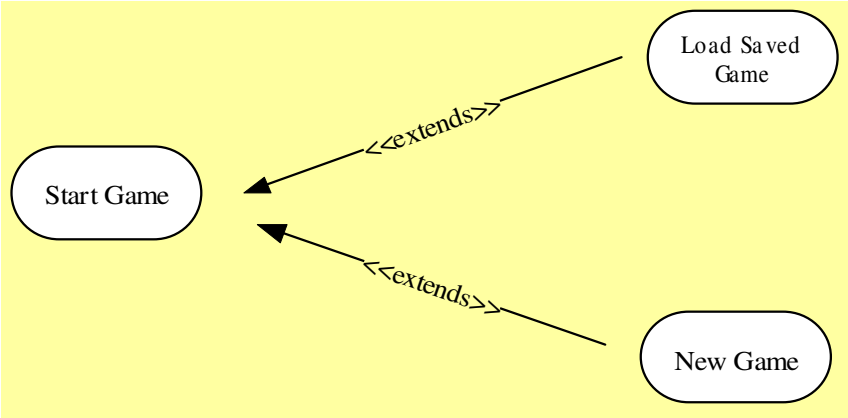
- There is a boundary for the maps of each level in the game.
- The walls are perpendicular to the floor.
- The game will be developed for windows.
- The game will be developed using C++ and OpenGL.
- The game will be developed in 5-8 months.

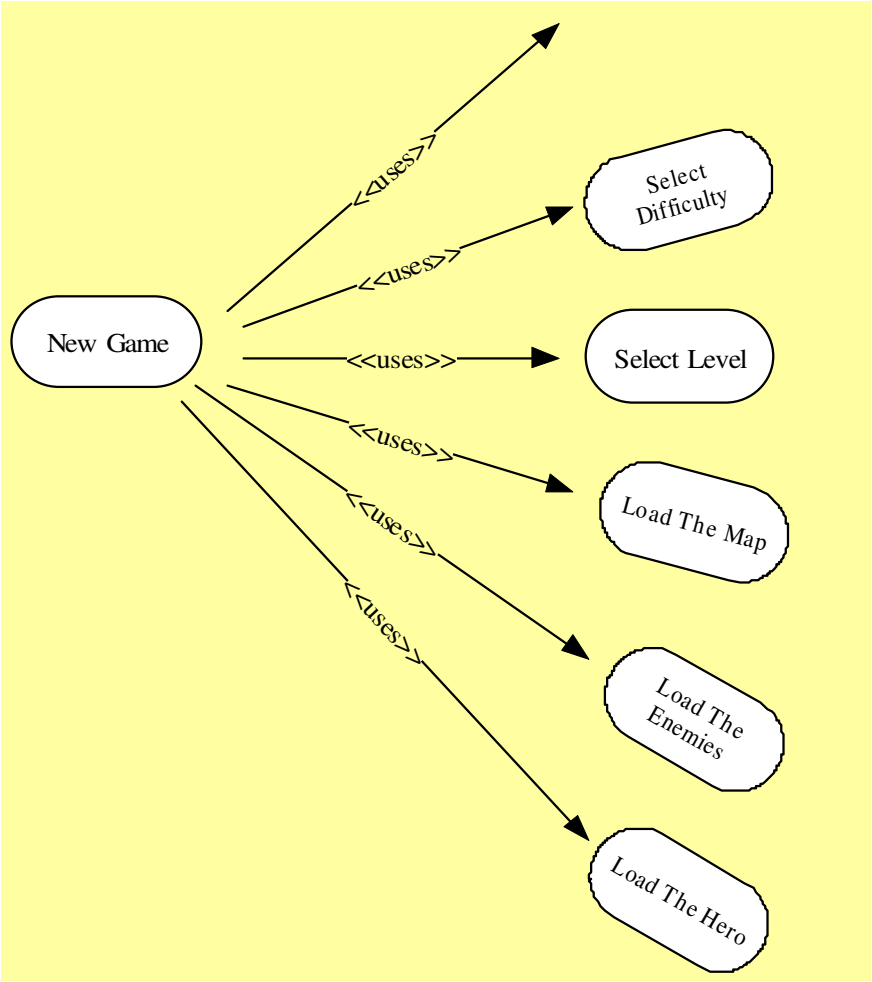
2.6 Operational Scenarios

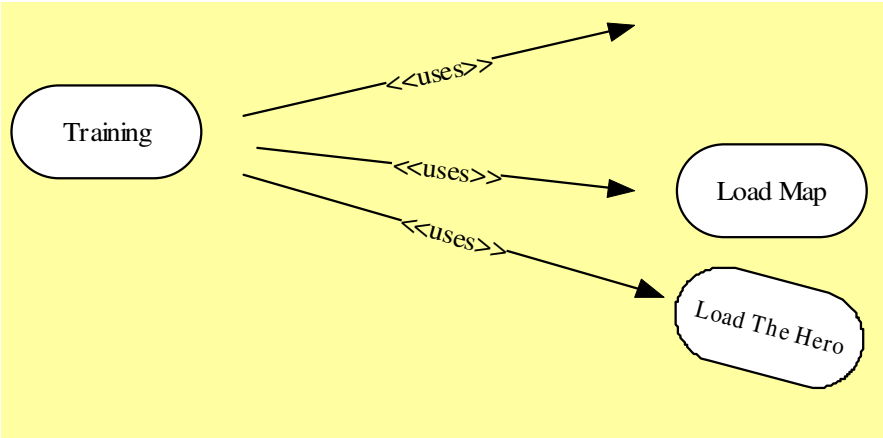
- Double Click the Game → Main Menu → Exit
- Double Click the Game → Main Menu → Gameplay
- Double Click the Game → Main Menu → Start New Game → Gameplay → Esc → Main Menu
- …→ Main Menu → Save Game → Main Menu → Gameplay
- …→ Main Menu → Training → Main Menu
- …→ Main Menu → Tutorial → Main Menu

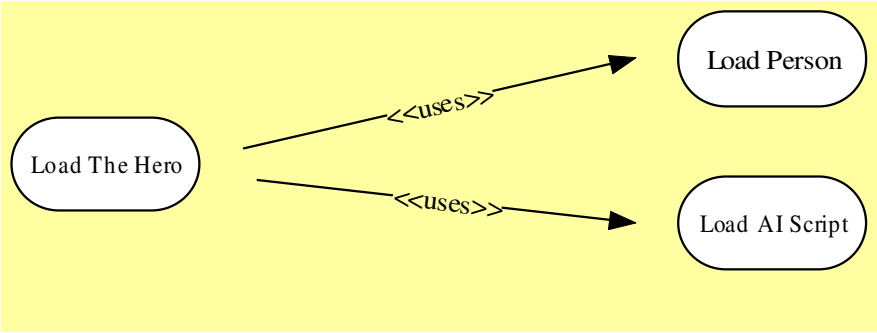
2.7 Use Case Diagrams

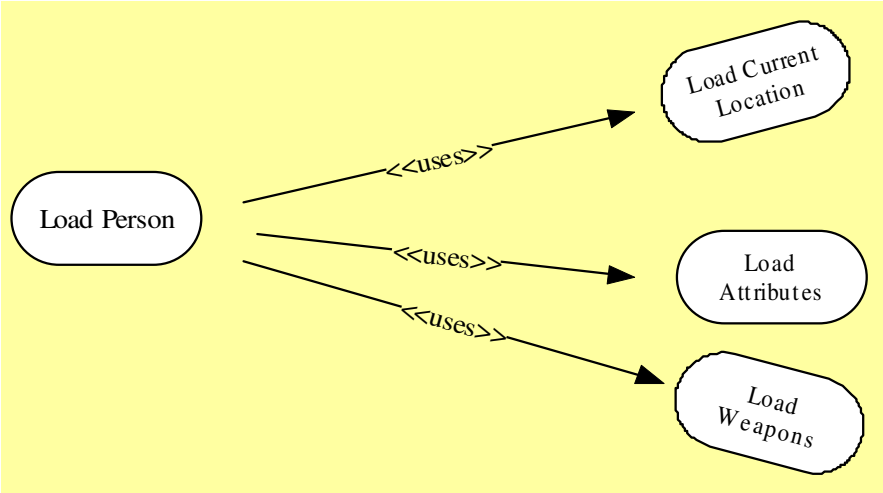


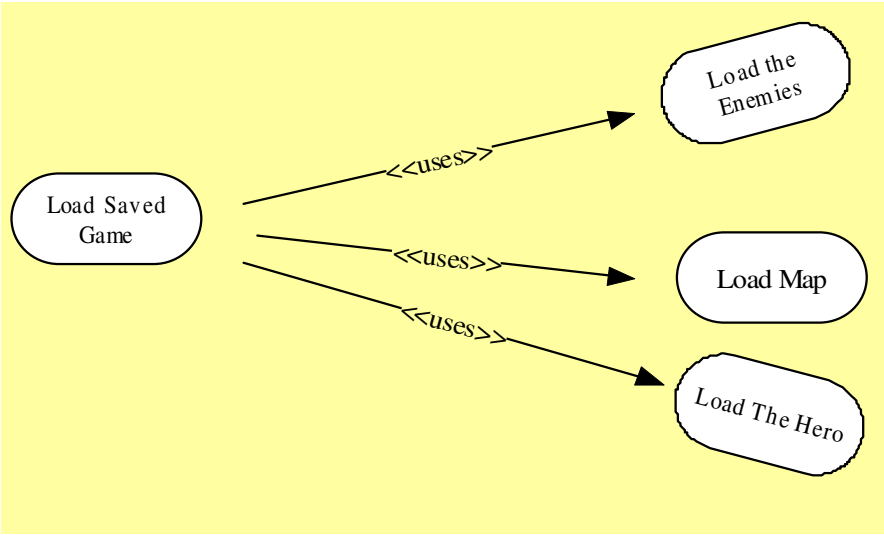

Load Saved Game
<<extends>>
Start Game
<<extends>>
New Game

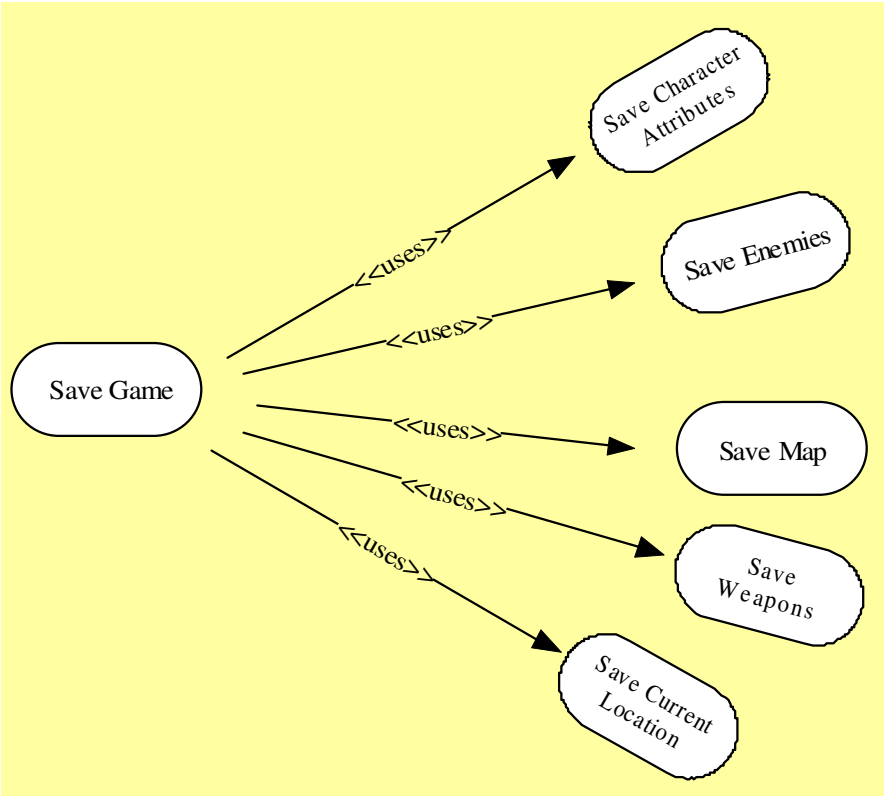

<<uses>>
Select Difficulty
<<uses>>
New Game
<<uses>>
Select Level
<<uses>>
Load The Map
<<uses>>
Load The Enemies
<<uses>>
Load The Hero


Training
<<uses>>
<<uses>>
<<uses>>
Load Map
Load The Hero


Load The Hero
<<uses>>
<<uses>>
Load Person
Load AI Script


Load Person
<<uses>>
<<uses>>
<<uses>>
Load Current Location
Load Attributes
Load Weapons


Load Saved Game
<<uses>>
<<uses>>
<<uses>>
Load the Enemies
Load Map
Load The Hero


Save Game
<<uses>>
<<uses>>
<<uses>>
<<uses>>
<<uses>>
Save Character Attributes
Save Enemies
Save Map
Save Weapons
Save Current Location

3. SYSTEM CAPABILITIES, CONDITIONS AND CONSTRAINTS

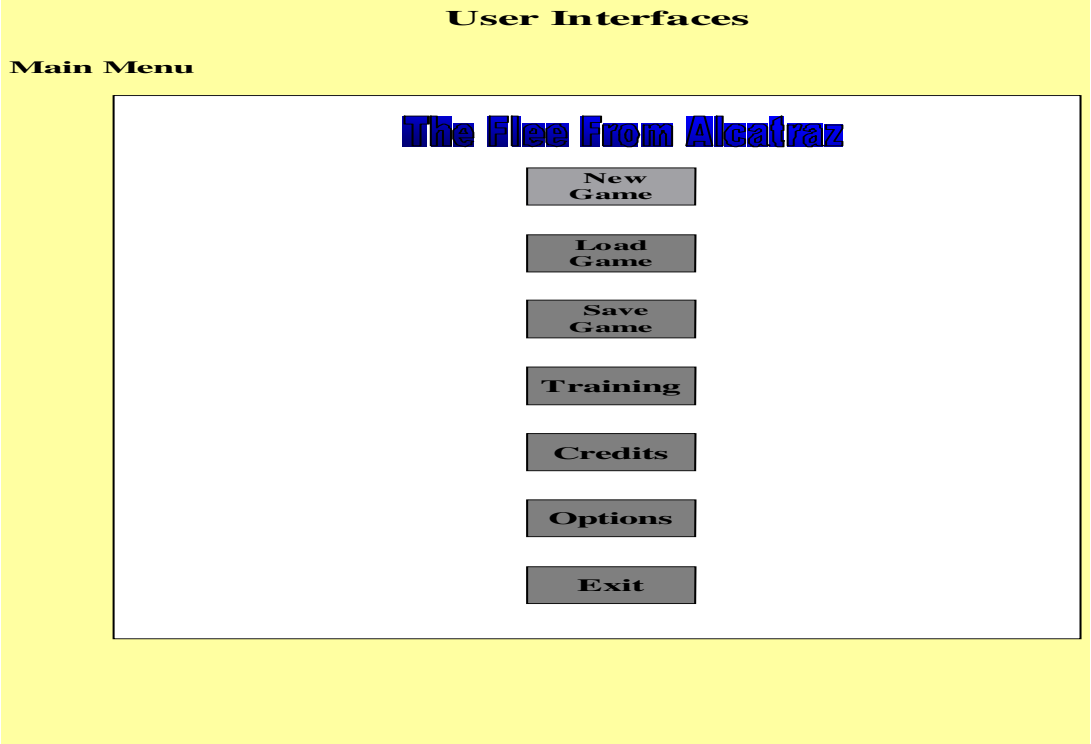
3.1 Environmental Conditions In Game

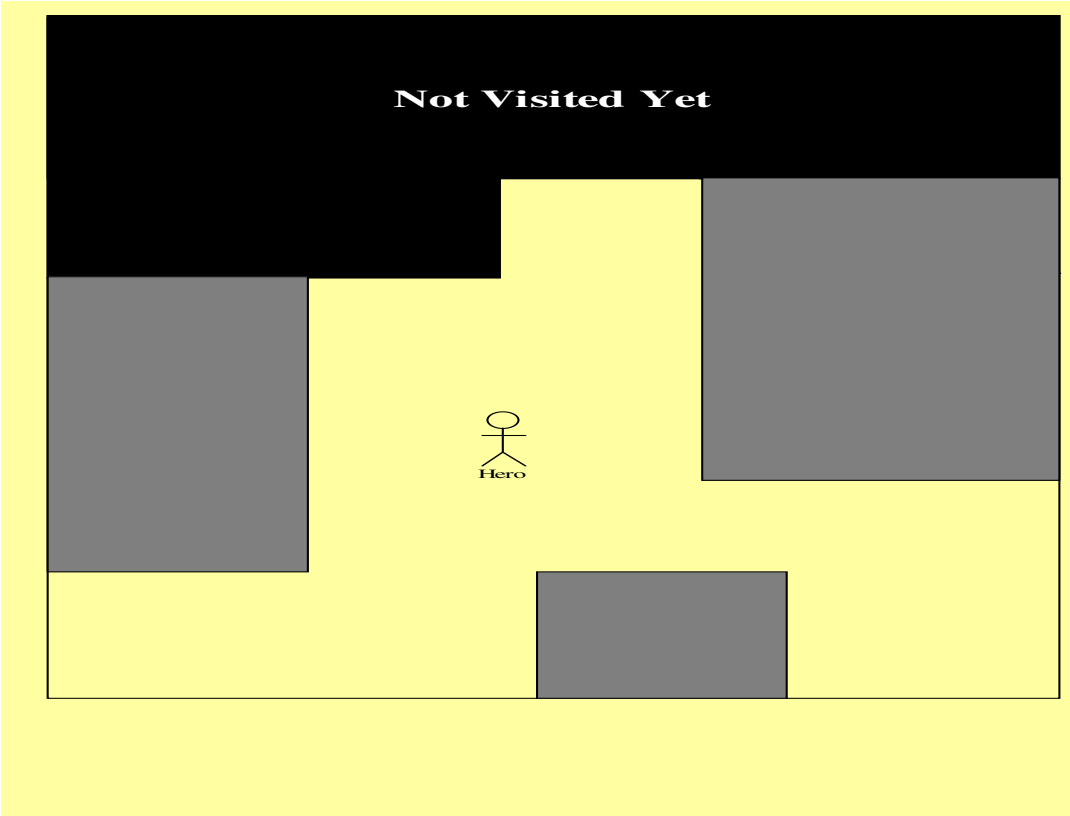
In the game there will be a lot of objects other than enemies. We will let the player to move some objects according to some calculations, depending on the character chosen. Since our game is a realistic game there are some implicit conditions such as there has to be gravity and collision detection implementation. i.e when our character jumps, he will fall down gradually.

3.2 Game Security

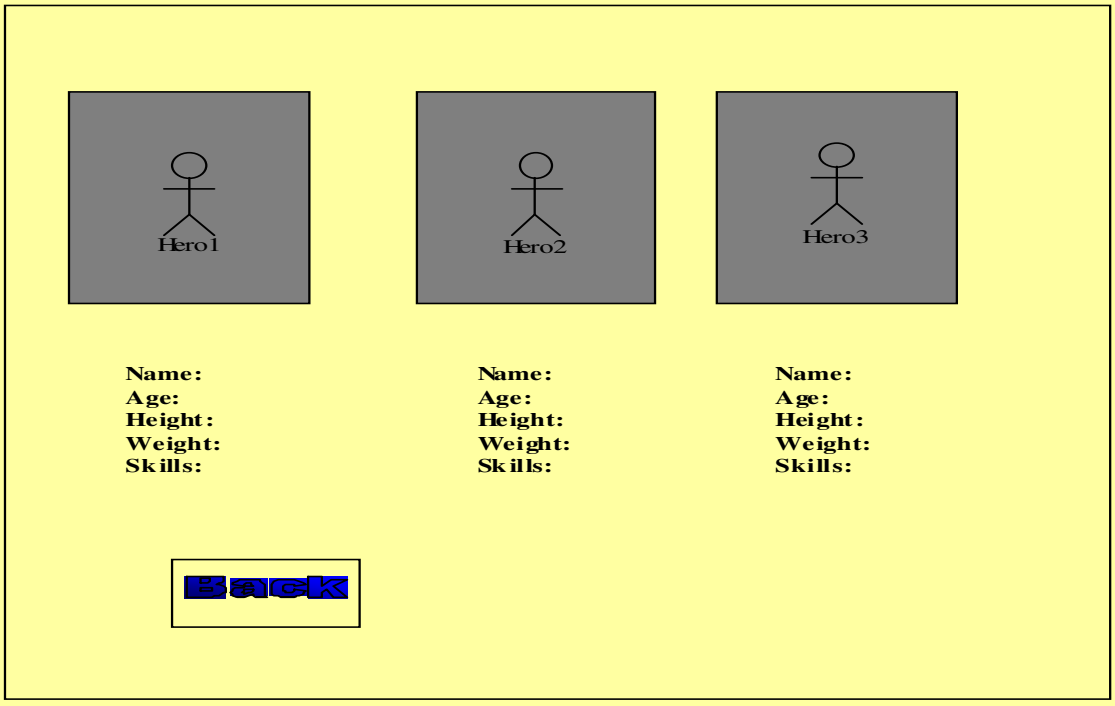
- When the user saves a game, he/she can put a password to avoid other people playing and changin his/her saved game.
- When the user presses “Alt + F4”, a confirmation window will appear to confirm whether the user wants to quit or return to game.
- When the user wants to save a game with an existing saved game name, name confirmation window will appear.
- During uninstallization, the user will be asked to whether he/she wants to uninstall everything or to keep saved games.

4. USER INTERFACES

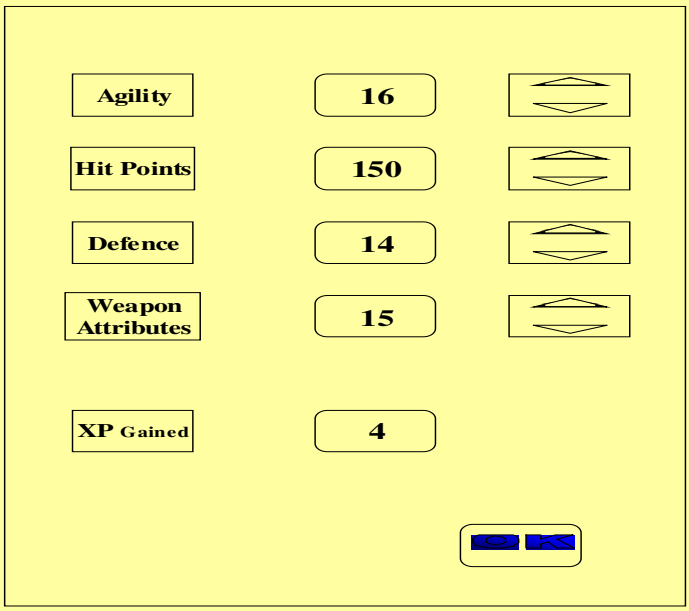




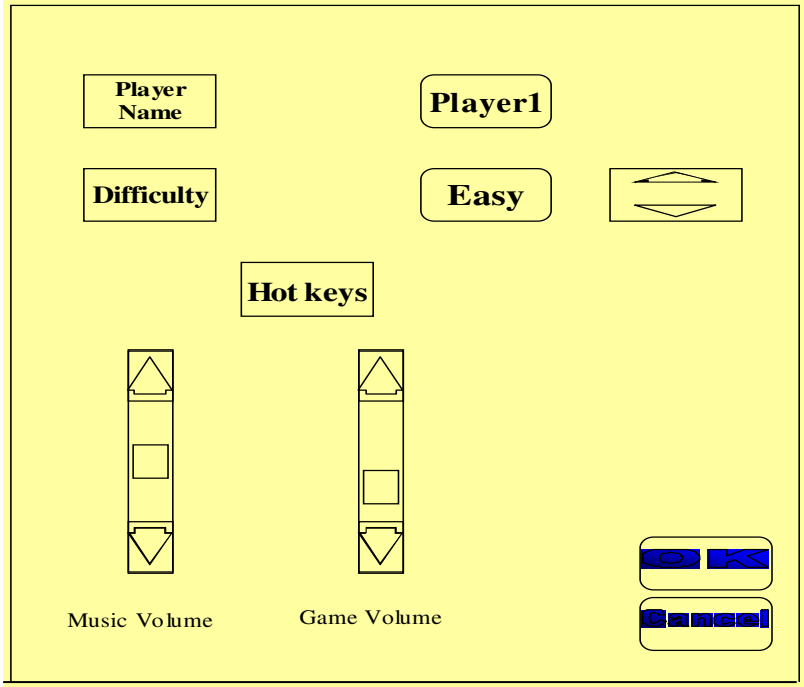
New Game/Training:



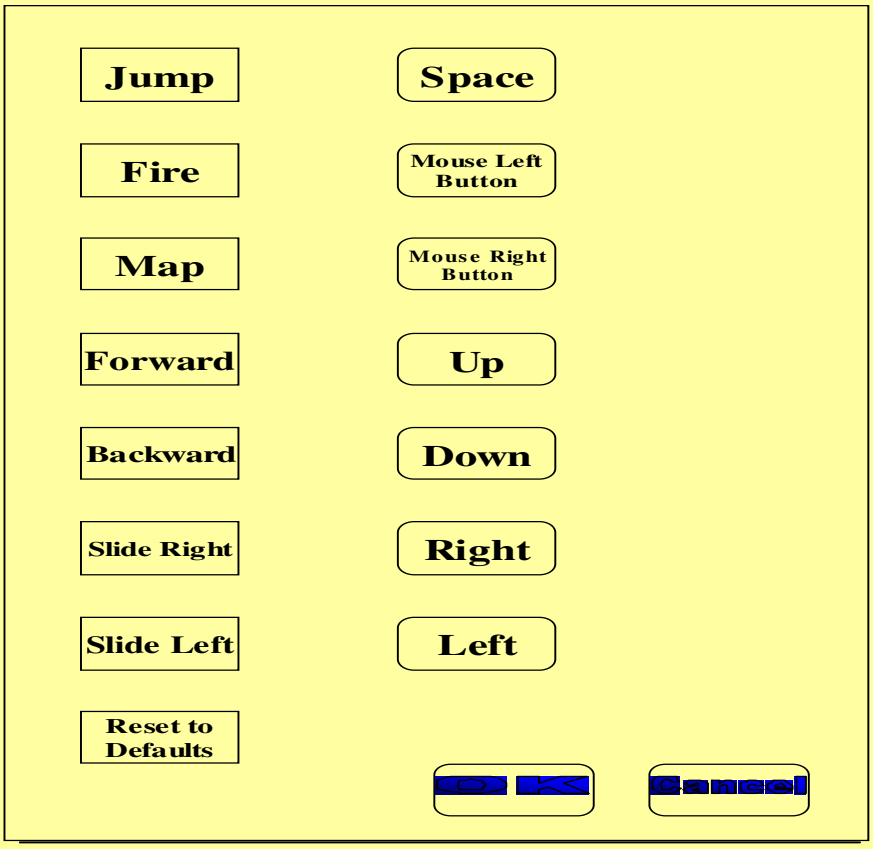
Attributes



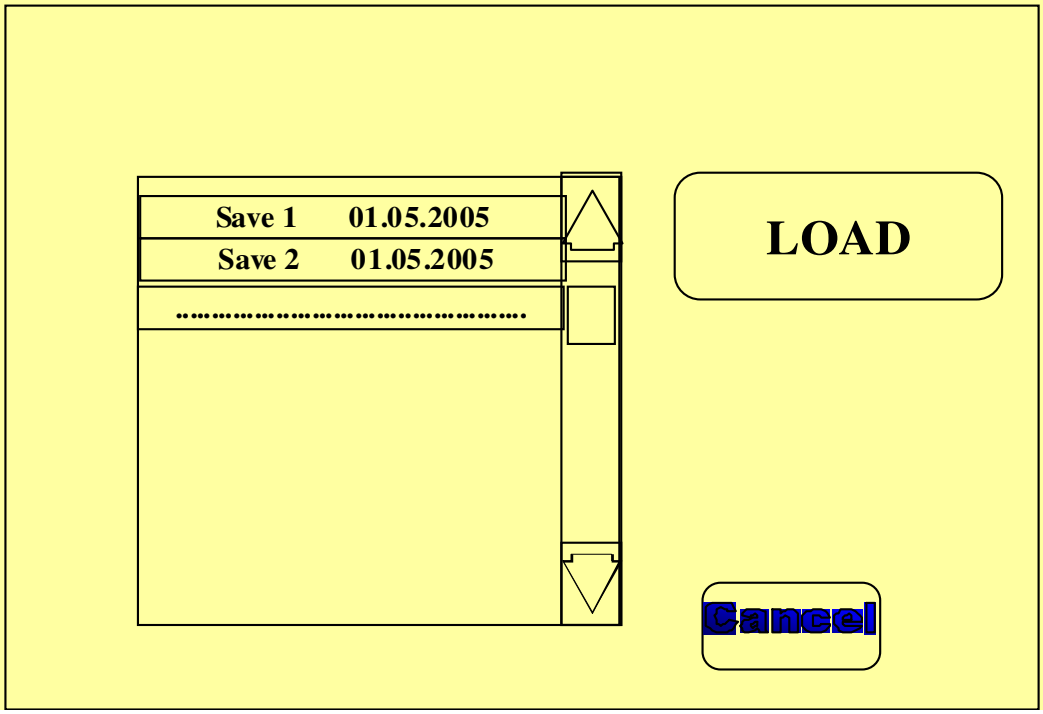
Options



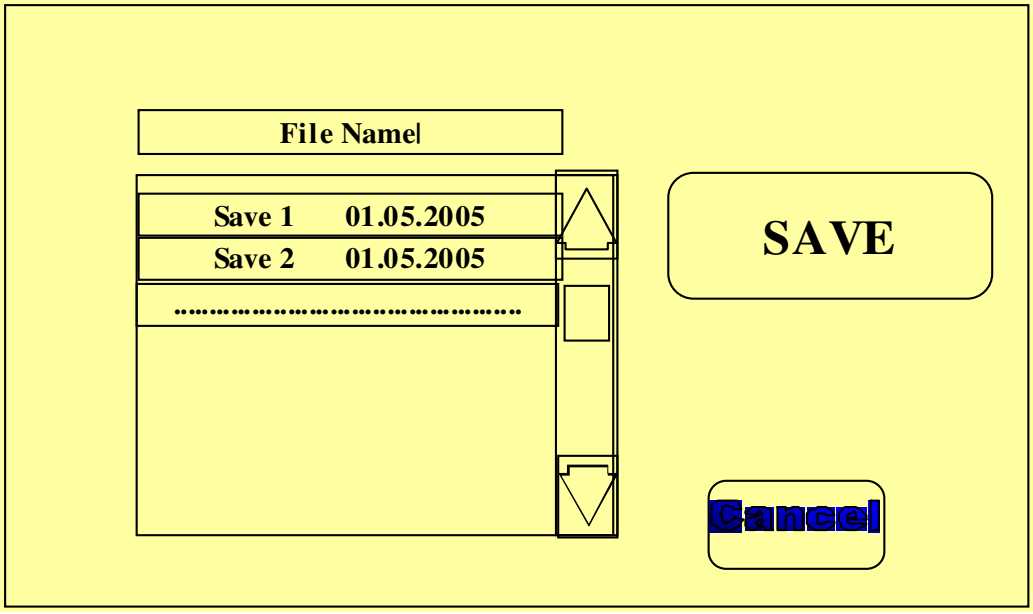
Hotkeys



Load:


Save 1 01.05.2005
Save 2 01.05.2005
LOAD
Cancel

Save:


File Name
Save 1 01.05.2005
Save 2 01.05.2005
SAVE
Cancel