

SOFTWARE REQUIREMENT SPECIFICATIONS

MAFIA

INTRODUCTION

The game Mafia is a single player game developed in c++ programming language using g++(ubuntu-linux) compiler. The main theme of the game is to place a bomb in opponent's (Artificial Intelligence) area and let it explode (the bomb explodes in three seconds) before the AI can diffuse it. Similarly, AI will try to do the same in player's area. A point is scored in favour of the one who explodes the bomb in opponent's area. Also a gun is available to both to shoot at each other resulting in decrement of the health bar associated with each player. The game is won by the one who kills the other guy and the no. of explosions done by him in the opponent's area is recorded as the total score.

The game uses various xpm format images for its GUI as well as for player movements. EzWindows is the platform used to display graphics and to run the entire game.

PROJECT OUTLINE

Main events

\$ Appearance of main screen with all button tabs (tabs refer to different menus like new game menu, credits menu etc.)

- New Game

|--> Start Game

|--> Back

- High Scores

Shows the highest score made by the player till now.

- Instructions

Shows controls of the player and theme of the game.

- Back

- Credits

Displays the name of all batch members, the batch TA as well as the name of the professor in charge of the course project (Prof. D. B. Phatak.).

- Exit

Closes the game window and returns to the terminal.

FUNCTIONS

MainScreen ()	Draws the background image of the main screen and all the other buttons.
ClearScreen ()	Clears all xpm images of the GUI
AI ()	Moves player 2 (AI) randomly
Explosion ()	Shows the animation of explosion
Timer ()	Shows the numbers for countdown and also calls Explosion function.
timer ()	Halts the program for a specific time given as argument by user.

STATUS OF COMPLETION

Unfortunately, due to certain logical problems, the game was not completed. The first problem was that the execution of loop of player 2 which is randomly controlled gradually slows down as time proceeds. This is because of re-initialization of floating point variables done again and again. Also timer function didn't work with multithreaded code. Hence, the remaining part of the game was also not completed.

IDEAS FOR FUTURE WORK

Basically, because of the autonomous nature of the AI or player 2, separate thread must be used for execution of player 2's loop so that main part of the game and player 2's loop execute parallel. Also, timer event must be used for motion of both player 2 as well as bullet. Usage of while or for loop won't work because they keep slowing down because of continuous changes in float variables. Yet some technique can be used either by using integer variables instead of float or executing loops in timer event. Different types of guns and different modes of

difficulty levels can be introduced for further development of this game. Theme of the game can also be changed as rescue of some person before he/she dies other than killing the opponent.

INDIVIDUAL CONTRIBUTIONS

HARDIK GODARA (TEAM LEADER)-Made most of the game including AI controlling, weapon handling, player 1 controlling and GUI of the game.

GANDHARV KANWAT-Wrote code for a little part of GUI of the game and did editing of all the player 1 and player 2 images in MS paint. Also edited background pages for credits and instructions.

HIMANSHU ARORA-Wrote code for movement of player 1 (consisting of four images) using arrow keys. Also, made Explosion and timer functions.

JYOTSNA GHAILAK-Wrote code for a little part of GUI of the game.

GOPAL GUPTA-Wrote code for sample movement of target box using arrow keys.

DINESH KUMAR-Made MainScreen and ClearScreen functions.

JAIDEEP POONIA-Did image editing for GUI buttons.

SYSTEM REQUIREMENTS

The game is specifically developed to run on UBUNTU operating system, which uses GCC compiler. The version of Ubuntu should be at least 9.04 or above. Also RAM must be at least 1 GB to avoid blinking of the player images. EzWindows library must be used as the game uses most of its classes and functionalities.

CREDITS

TEAM MEMBERS

1. HARDIK GODARA(TEAM LEADER)
2. GANDHARV KANWAT
3. HIMANSHU ARORA

4. JYOTSNA GHAILAK

5. GOPAL GUPTA

6. DINESH KUMAR

7. JAIDEEP POONIA

BATCH TA-PRAMOD MINJ

COURSE PROFESSOR-DR. D. B. PHATAK

