Bilkent University

Department Of Computer Engineering

Object Oriented Software Engineering Project

Academic Warfare : The Conflict in Bilkent

Fınal Report

Doğukan Yiğit Polat, Selin Fildiş, Yasin Erdoğdu, Onur Elbirlik

Instructor: Uğur Doğrusöz

Design Report

Dec 24, 2016

Contents

Introduction and Project Summary

The project is an implementation of a Tower Defense game. Tower Defense games mostly consists of enemies coming to invade the player's base and the player is trying to defend their base via weapons and/or defenses. The enemies come in waves in these types of games. Our version is called Academic Warfare: Conflict in Bilkent. The player (aka the student) tries to prevent the enemies (the academics) from completing the map. When the map is completed by the academics the student is defeated. But the students have the opportunity to defend themselves via weapons they can buy and bonuses which will be given out the game. The weapons are unlocked and bought with coins they gain throughout the game and the bonuses are randomly distributed throughout the game. There are 4 types of weapons which are called: Tekman's Cannon the basic weapon, Oktel's Double Trouble, Bilka's Bazlamaya Ayvalık Spitter and Doğramacı's Railgun which are unlocked with coins in the game. The weapons are also bought for coins. Also, there are 4 types of bonuses which are life steal which steals life from the enemies, thunderbolts and mines which deal damage and freezer which freezes the enemies for 5 seconds. The game ends when the user quits or is defeated.

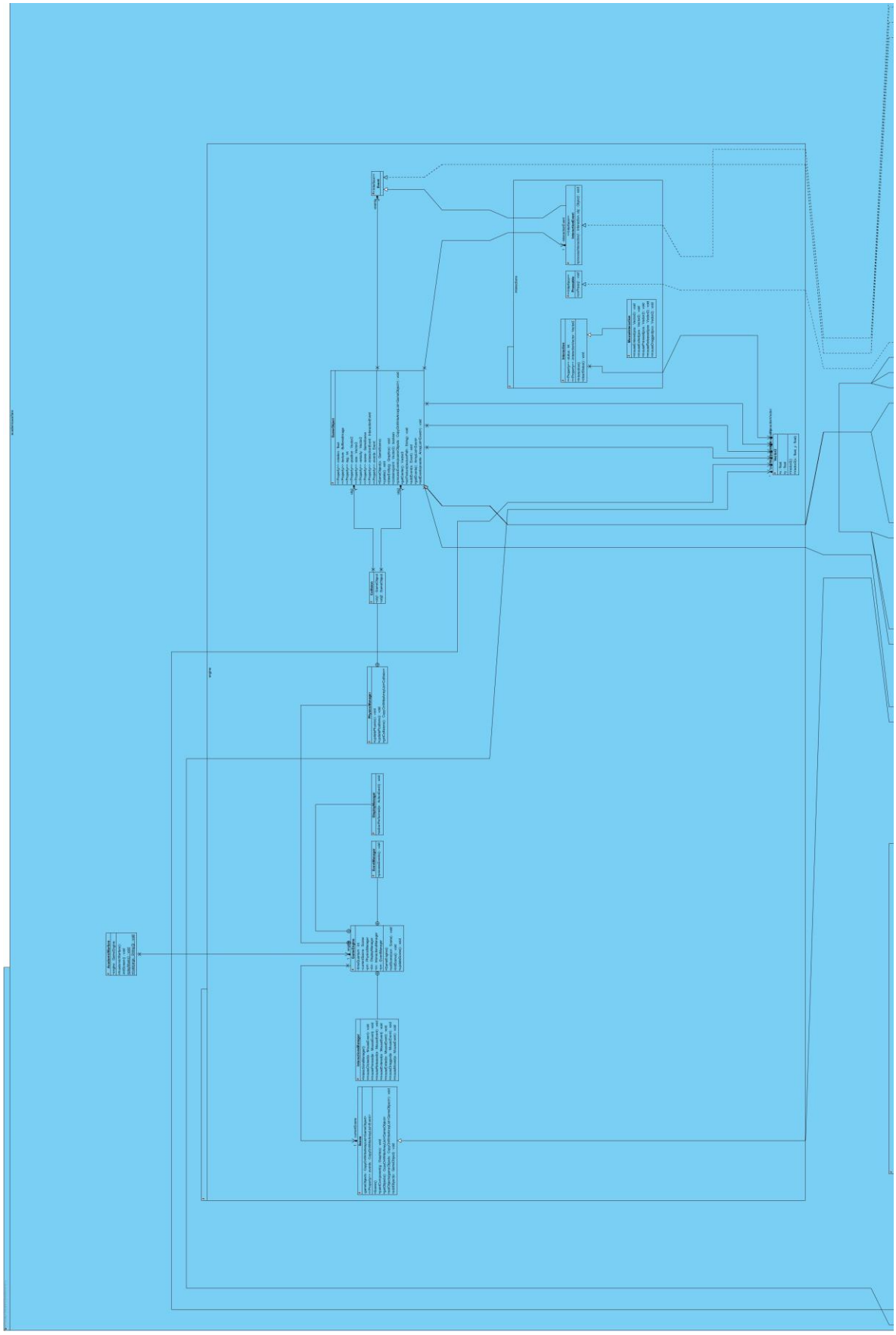
Installation

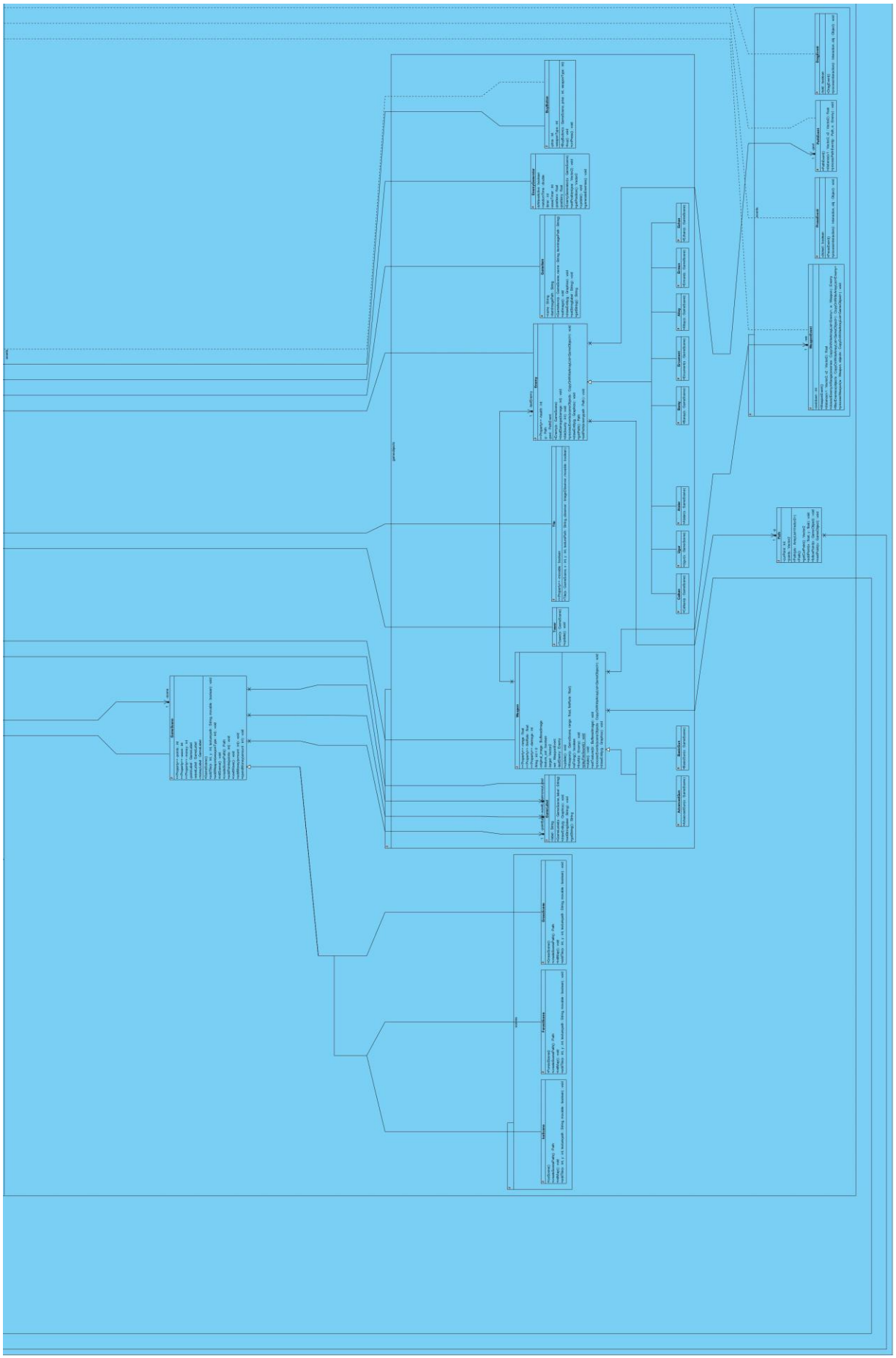
The Project will require a Java Runtime Environment installed on the computer to run.
To run simply click on the ".jar" extended file to run. The JRE (Java Runtime Environment) will start up the program on the computer. Since JRE runs on multi-platforms, the game will run on all operating systems which have the environment installed.

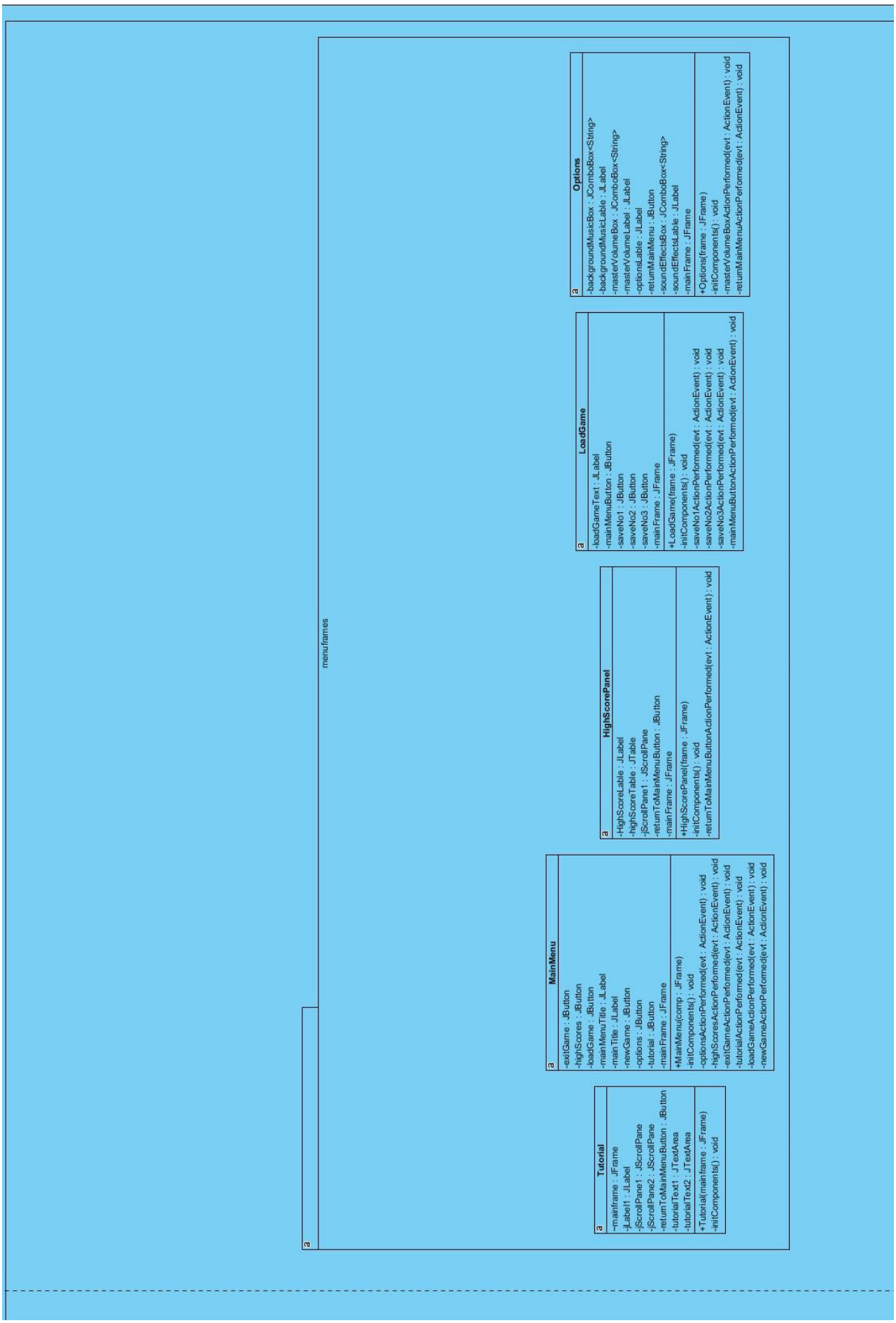
Implementation Process

The project has tried to be implemented according to the design document but some unpredicted circumstances there might be changes. The latest class diagram and the changes in the design are stated below.

Project UML






menuframes
Options
-backgroundMusicBox : JComboBox<String>
-backgroundMusicLable : JLabel
-masterVolumeBox : JComboBox<String>
-masterVolumeLabel : JLabel
-optionsLable : JLabel
-returnMainMenu : JButton
-soundEffectsBox : JComboBox<String>
-soundEffectsLable : JLabel
-mainFrame : JFrame
+Options(frame : JFrame)
-initComponents() : void
-masterVolumeBoxActionPerformed(evt : ActionEvent) : void
-returnMainMenuActionPerformed(evt : ActionEvent) : void
LoadGame
-loadGameText : JLabel
-mainMenuButton : JButton
-saveNo1 : JButton
-saveNo2 : JButton
-saveNo3 : JButton
-mainFrame : JFrame
+LoadGame(frame : JFrame)
-initComponents() : void
-saveNo1ActionPerformed(evt : ActionEvent) : void
-saveNo2ActionPerformed(evt : ActionEvent) : void
-saveNo3ActionPerformed(evt : ActionEvent) : void
-mainMenuButtonActionPerformed(evt : ActionEvent) : void
HighScorePanel
-HighScoreLable : JLabel
-highScoreTable : JTable
-jScrollPane1 : JScrollPane
-returnToMainMenuButton : JButton
-mainFrame : JFrame
+HighScorePanel(frame : JFrame)
-initComponents() : void
-returnToMainMenuButtonActionPerformed(evt : ActionEvent) : void
MainMenu
-exitGame : JButton
-highScores : JButton
-loadGame : JButton
-mainMenuTitle : JLabel
-mainTitle : JLabel
-newGame : JButton
-options : JButton
-tutorial : JButton
-mainFrame : JFrame
+MainMenu(comp : JFrame)
-initComponents() : void
-optionsActionPerformed(evt : ActionEvent) : void
-highScoresActionPerformed(evt : ActionEvent) : void
-exitGameActionPerformed(evt : ActionEvent) : void
-tutorialActionPerformed(evt : ActionEvent) : void
-loadGameActionPerformed(evt : ActionEvent) : void
-newGameActionPerformed(evt : ActionEvent) : void
Tutorial
~mainframe : JFrame
-jLabel1 : JLabel
-jScrollPane1 : JScrollPane
-jScrollPane2 : JScrollPane
-returnToMainMenuButton : JButton
-tutorialText1 : JTextArea
-tutorialText2 : JTextArea
+Tutorial(mainframe : JFrame)
-initComponents() : void

Changes to the Analysis and Design Documents

The design document, although predicted well, required some unpredicted changes which are stated bellow.

Changes to the Analysis Document

Load and Save Game functions

The load and save functions could not be implemented due to timing issues.

In-game Load – Save- Options Menu

This menu wasn't implemented as well due to timing issues.

Towers

Towers are excluded from the game.

Changes to the design documents

Enemy's child classes

These classes were added to make implementing the Academics easier. The Enemy Class has 8 child classes for each Academic.

Scene's child classes

Each map is generated as a child class of scene.

Packaging

The Packaging was changed to optimize and organize the code.

Methods

Some methods were changed to meet needs during implementation.

Save and Load Game

Although we were supposed to save game and record game saves as a binary file using the "Serializable" interface, we are unable to accomplish this task because there is not enough time anymore. There is no load option since, there is not any saved game.

Pause and Resume

There are not pause and resume option for game playing. Therefore, player are not able to pause the game and resume the game afterwards.

Ready

We were supposed that after the player select a new game option, s/he had to be click "Ready" button to start game playing. Unfortunately, due to inadequate time, we did not fulfill this task. So, player can start to play game after s/he selects a new game.

Power-ups

We were planning to there will be four different power-ups in the game in order to provide user with extra changes to defeat enemy. However, there is not any power-ups for user in the current version of the game. We were unable to finish this option before the deadline of project demos.

Enemies

We supposed that there will be academics and assistants enemies in the game. There are academics as enemy in the current version of the game, however, we did not include the assistants as enemies in the game. Moreover, we designed the enemies such that enemies destroy the tower with firing bullets. Hence, in the implemented game, enemies destroy tower just by arriving to the tower location. In addition to this, some changes about enemy health and velocity were required to make the game more maintainability and playable.

Weapons

We were planning to have four different weapons to defeat enemies in the game. Yet, there are not four different weapons in the game. We have just only some academics as enemies in the game. In addition to this, some changes about weapon damage and frequency were required to make the game more maintainability and playable.

HighScore

We supposed that the game will keep top 10 scores with player name information. User will be able to see top 10 high scores of the game via clicking “High Score” button. However, we did not also handle the recording and viewing high scores.

Exposed Documentation

The user's manual is given below.

Introduction:

Academic Warfare is a tower defense game which consists of different weapons, enemies and 3 maps. Game is implemented in a wave system. Players should protect their towers until the wave ends. The waves will continue as long as the player keeps on playing and after each wave the player will gain coins according to the number of enemies eliminated in that particular wave. The game has 3 different maps and the second and third maps will unlock if the user completes the waves in the previous map successfully. The user can choose which map they want to play with after unlock. The game can be paused, saved and continued. The saved game can be loaded in the beginning as well.
The game will be a desktop application written in java and will use a mouse for the user to interact with the game.

System Requirements and Installations:

System Requirements:

The Standard Java Runtime Environment (JRE) must be installed before installing the game. To download: http://java.sun.com/

Minimum system requirements:

- Windows XP/7
- Pentium2 233 MHz CPU or higher.
- 512 MB of RAM or higher.
- Screen resolution: 1024x768
- Integrated graphics card (e.g. laptops are compatible.).

Recommended system requirements:

- Intel Pentium III 800MHz or equivalent processor (Athlon/Duron/Celeron).
- 512 MB of RAM or higher.
- Screen resolution: 1366x768.
- SoundBlaster compatible sound card.
- For better performance, separate graphics card. (NVIDIA GeForce 3 or ATI Radeon 8500 video card)
- SUPPORTED CHIPSETS: NVIDIA - GeForce 6800, GeForce FX (5950, 5900, 5800, 5600, 5200), GeForce4 Ti, GeForce3 ATI - Radeon x800, 9800, 9700, 9600, 9500, 9200, 9000, 8500.

Installations:

Unzip the academicWarfare.zip to a directory on your computer and to run the game you have 2 choices.

1- Run the academicWarfare.jar to execute the game(which it is recommended)

2- If you have knowledge of java programming, by running our code in java and can play the game.

Playing the Game:

More details about the game and the gameplay, list of weapons, maps will be stated in this section.

Overview of the Game:

This tower defense game is implements a wave system. The wave system is groups of enemies entering and moving like a wave such as group 2 enters 10 seconds after group 1. The waves will continue as long as the player keeps on playing and after each wave the player will gain coins according to the number of enemies eliminated in that particular wave.
Thus, the coins earned will increase as the game continues as the waves will get harder. To defeat these waves the player will build weapons to stop their enemies from reaching and destroying their base. The game will also have power-up's and the user will collect these
power-up's to help them defeat their enemies.

The game has 3 different maps and the second and third maps will unlock if the user completes the waves in the previous map successfully. The user can choose which map they want to play with after unlock.

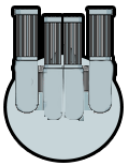
List of Weapons

There are four weapons that the user can choose from, some of these weapons can be unlocked in-game as the player goes further and bought the coins the user gains after each wave. Weapons fire to the nearest enemy.

Tekman's Cannon: Tekman's cannon is the base weapon in this game, the cannon shoots with 1 ball per second and deals 10 damage. To place the gun the user has to pay $20.

Oktel's Double Trouble: This gun is one of the unlockable guns in the game, can be unlocked for $200. Deals 7 damage but its fire rate is 2 shots per second. So it is a much better weapon than Tekman's Cannon. Build cost of Oktel's Double Trouble, after unlock, is $150.

Doğramacı's Rail Gun: This is the most powerful weapon in the game. The gun deals 20 damage and fires at rate 5 shots per second and the build cost is $350 after unlock. The unlock fee is $2000.



Bilka's Bazlamaya Ayvalık Spitter: This weapon slows down the target by %10 on its each shot and shoots at rate 5 shots per second like the Doğramacı's Railgun. The unlock fee is $250 and in-game build cost is $150.

Enemies

There will be an academic and assistants accompanying the academic in every wave. In further waves the possibility of the department chair, the dean of engineering and the rector appearing as an academic will increase.

Uğur Doğrusöz

Abdullah Atalar

Mehmet Baray

Can Alkan

Ercüment Çiçek

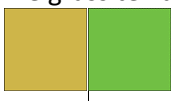
Erman Ayday

Ezhan Karasan

List of Maps

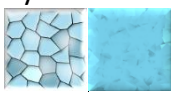
Grass Terrain

The grass terrain is the basic terrain type. It does not have any affect on neither the enemies.



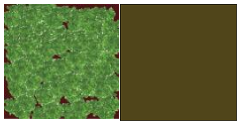
Snow Terrain

The Snow terrain's road is icy and is hard for the enemies to move, so their speed is dropped by %10 in this terrain.



Forest Terrain

The forest terrain is similar to the snow terrain but in reverse. In this terrain, the speed of the enemies are increased by %10.



Power-ups

In the game, there are some bonuses to help out the player. There are four bonuses which are life steal, freeze and thunder bonus.

Thunder Bonus: This bonus enables user to explode each grid cell with probability %10, so if enemy is inside the cell that has exploded then it will die.

Life Steal: With this bonus, user can increase tower health by %1 for each enemy killed in during 5 seconds from its activation.

Freeze: User can freeze all enemies for 5 seconds by using this bonus.

Mines: User can locate this bonus to any cell in the game screen to destroy enemies within its cell.

Score:

User can able to see current his/her own score and wave number at any time during the game.

Health:

User can see his remaining health at the right corner of the game screen with the score. Health will be different according to the enemies.

Controller:

Default Mouse Controls:

UP	when user drags the mouse forward
DOWN	when user drags the mouse backward
LEFT	when user drags the mouse left
RIGHT	when user drags the mouse right
PLACE THE WEAPON	when user clicks the left mouse button

Save and Load:

User will be able to save current state of game and user can load any saved games in order to continue to play where they left off.

Pause and Resume:

Player will be able to pause the game and resume the game afterwards.

Menus:

Main Menu:

The main menu consists of buttons “New Game”, “Load Game”, “Tutorial”, “Options”, “High Scores”, “Exit” which navigates the user to the desired point.

Load Game:

The main menu consists of buttons “New Game”, “Load Game”, “Tutorial”, “Options”, “High Scores”, “Exit” which navigates the user to the desired point.
In the load game screen as shown, the user can select 3 of his previously saved game, and continue to play the saved game or he can return back to the main menu

High score Table

The user can view his/her high scores through this table shown in figure 11. Afterwards, the user can return to the main menu.

Options

The user can change sound options through this menu shown in figure 12 and return to the main menu afterwards.

Tutorial-Help

The user can view the tutorial whose template as shown is to understand how the game works. The tutorial will consist of multiple pages.

Gameplay

The template for the game play screen is shown. In the larger square the game will be played while in the right squares the available weapons, list of used bonuses and stats regarding the wave will be displayed. Also there is a small menu on the upper left corner in which the user can return to the tutorial page, save game, pause and change options.

Warranties and Troubleshooting:

“Academic Warfare” team hopes you will have fun during the game. However, it is not guaranteed because the term “fun” can change from person to person. If you face with

problems or you had trouble while playing the game, please feel free to contact us by sending us e-mail.

yigit.polat@ug.bilkent.edu.tr
yasin.erdogdu@ug.bilkent.edu.tr
onur.elbirlik@ug.bilkent.edu.tr
selin.fildis @ug.bilkent.edu.tr

Technical Support

If you are having problems running the game, you may want to try updating your hardware drivers before contacting technical support:

ATI Technologies
http://www.atitech.com/

NVIDIA
http://www.nvidia.com/

Creative Labs
http://www.creativelabs.com/

HERCULES
http://www.hercules.com/

Turtle Beach
http://www.turtlebeach.com/