Bilkent University

Department of Computer Engineering

CS319 Object-Oriented Software Engineering

Astunc: Bullet Hell Game

Analysis Report

Project Group 12: Orhun Kar, Burak Şenel, M. Caner Çalışkaner

Supervisor: Uğur Doğrusöz

Feb 20, 2015

Contents

1. INTRODUCTION 1

2. SYSTEM OVERVIEW 1

2.1. GAMEPLAY 1
2.2. STAGE 1
2.3. PLAYER 1
2.3.1. MULTIPLAYER 1
2.4. ENEMIES 1
2.4.1. NORMAL ENEMIES 1
2.4.2. BOSS ENEMIES 1
2.5. POWER UPS 1
2.5.1. DAMAGE BOOST 2
2.5.2. TIME BONUS 2
2.5.3. SHIELD 2

3. FUNCTIONAL REQUIREMENTS 2

3.1. PLAY GAME 2
3.1.1. MULTIPLAYER GAME 2
3.2. TUTORIAL 2
3.3. CHANGE SETTINGS 2
3.4. HIGH SCORES 2
3.5. PAUSE GAME 2

4. NONFUNCTIONAL REQUIREMENTS 2

4.1. GAME VISUALS 2
4.2. GAME PERFORMANCE 2
4.2.1. GRAPHICAL PERFORMANCE 2
4.2.2. MULTIPLE INPUT PERFORMANCE 2
4.2.3. HIT BOX DETECTION PERFORMANCE 3
4.3. USER-FRIENDLY INTERFACE 3
4.4. EXTENDIBILITY 3

5. SYSTEM MODEL 3

5.1. USE CASE MODEL 3
5.1.1. PLAY GAME 4
5.1.2. TUTORIAL 4
5.1.3. CHANGE SETTINGS 5
5.1.4. HIGH SCORE 7

6. CONCLUSION 7

Analysis Report

Astunc: Bullet Hell Game

1. Introduction

We decided to develop a game called Astunc. It will be a bullet hell type of game. Bullet hell is a subgenre of shoot them up games. The main objective is the destroy the enemy entities on screen with the projectiles from user. Usually these games are set on space. We can describe this kind of game as Chicken Invaders with more projectiles on screen that limits the movement of the player.

2. System Overview

2.1. Gameplay

To separate Astunc, from the other space shoot them up bullet hell games, we decided to based it on time rather than points gained by destroyed enemies. The shortest time will act like the highest score. The purpose of Astunc will be same as the other these type of games. The player will try to avoid the gigantic numbers of enemy projectiles called "Bullet Hell" and destroy the enemy entities that are creating the bullet hell. After some number of normal enemy entities, there will be boss enemy entity which will spawn more enemy projectiles than normal enemy and will take more hits to destroy.

2.2. Stage

The game will be divided into 5 stages, each stage will harder and harder to play as player progresses in game. Every stage will consist of numerous normal enemies and a boss enemy which will be explained in section 2.4.
Each stage will have a planet theme like earth, mars *etc.* In each stage player will face the enemy types that are inclusive to the theme.

2.3. Player

Player will be presented as a spaceship on screen, the spaceship will be able to shoot projectiles that have blue glow. The projectiles will be able hit and destroy normal enemies in two shots unless player got a random power up that will be explained in section 2.5. Player's objective is to destroy every enemy entity.

2.3.1. Multiplayer

If there is up to 3 players, they can play together. Every specification of player applies to all of the players but the spaceships image and color will be different from each other to differentiate the players.

2.4. Enemies

Enemies are the computer controlled and as explained earlier aim of the game is to destroy all the enemies. Generally, enemies will move in a certain path and will shoot gigantic number of projectiles

2.4.1. Normal Enemies

Normal enemies are encountered from the start of a stage until the boss enemy is encountered. Normal enemies are slightly different depending on the stage player is on. Player is able to destroy each normal enemy in two successful shots, unless a power up is active which enables the player to destroy normal enemies in one hit.

2.4.2. Boss Enemies

Boss enemies are encountered after all of the normal enemies in a stage are defeated. Boss enemies are larger and take more than two shots to take down and have bigger and more complex projectile patterns. Defeating a boss enemy means that a stage is completed and player can move on to the next stage.

2.5. Power Ups

Power ups are collectibles that will appear in stages and when collected have different positive effects depending on the type of power up collected. Power ups that are not collected within a time period will disappear.

2.5.1. Damage Boost

Damage boost power up when collected causes player projectiles to deal more damage to enemies. While this power up is active player projectiles will kill normal enemies in one hit.

2.5.2. Time Bonus

Time bonus power up when collected will reduce the timer by a set amount. Since time is used as the score this power up acts as a score bonus.

2.5.3. Shield

Shield power up when collected will protect the player from enemy projectiles for a certain amount of time.

3. Functional Requirements

3.1. Play Game

If it is first time the player plays the game, after pressing the button "Play Game" Astunc will ask to player if he/she wants to play the tutorial or not. If the player chooses to play tutorial, he/she will be faced with an introductory gameplay. If not than the game will start immediately.
If it is not the first time the player wanted to play the game, it will start the game immediately.

3.1.1. Multiplayer Game

While a player plays the game if there is another player who wants to join, they can press key bind for join action for it. The key bindings will be explained in section 3.3. After a player joins he or she will have the average of time the other players have to prevent getting unfair high scores that will be explained in section 3.4.

3.2. Tutorial

On an occasion the user wants to see the tutorial again, such as another person is trying game on a system that the game is played on like a friend's house, there will be option for replay introductory gameplay.

3.3. Change Settings

If the user presses the "Settings" button, they will have the options the change such as the volume of the game, the image of spaceship, change the player name that will be showed in high scores and key bindings for movement and attacks of each player.

3.4. High Scores

Scoring in Astunc is very straight forward. There is no way to gain score in this game. Ranking will be determined with the time spent on the game. So lowest time will be displayed at the top of the ranking system. With this type of scoring system Astunc will have the replay ability. In other words, players will have a reason to play the game again after beating it.

3.5. Pause Game

Game will be paused by escape key. When this key is pressed a big string which is "PAUSED" will be displayed at the middle of the screen. After this point pressing the same key will reactivate the game but a counter from 3 will be shown in order to give player some time to prepare him/herself

4. Nonfunctional Requirements

4.1. Game Visuals

Game Visuals must be easy to understand. Such as which ships are friendly, which are enemy and the projectiles must be visible while they are on screen, they should not blend in with the background.

4.2. Game Performance

4.2.1. Graphical Performance

The game must be playable with a reasonable screen refresh rate which enables players to react enemy ships and the enemy projectiles in time. Since there will be a lots of projectile the number of projectiles in screen should not limit the playability by not affecting the animations and screen refresh rate.

4.2.2. Multiple Input Performance

Since there will be up to 3 players and each player can press multiple number of buttons at the same time, the key press detection must be precise enough to get the inputs from each player without any latency.

4.2.3. Hit Box Detection Performance

The main objective of the bullet hell genre of games is to avoid touching the enemy projectiles, so hit box detection must be precise and it must be without any errors to maximize the player satisfaction.

4.3. User-Friendly Interface

The interface of menus and screen must be easily understandable and clean to make it easy to navigate through. The in-game user interface that shows necessary information about the player and the stage must be understandable by a look from corner of the eye since the player will be and should be focused on the game and the enemy projectiles.

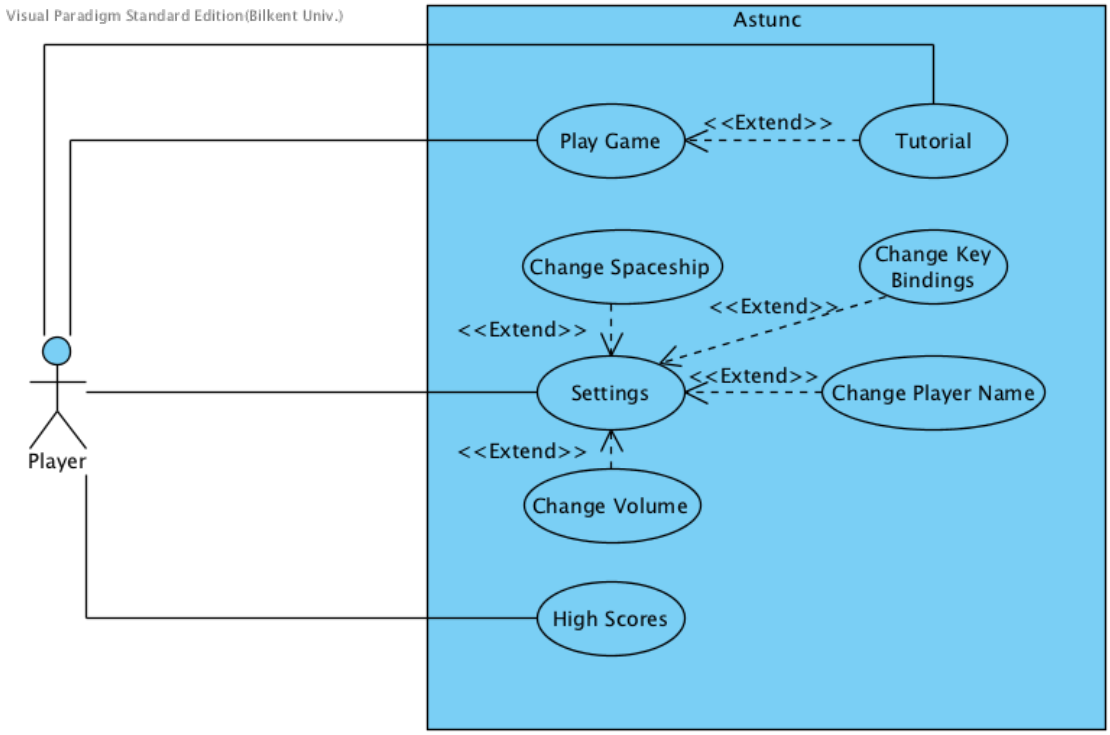
4.4. Extendibility

As a good software engineering practice and to appeal possible customers, the game should be extendable. Possible extensions listed below:

- Gamepad Support
- Mouse Support
- Adding 4th player to Multiplayer Game
- Online Multiplayer Game
- New Enemies
- New Bosses
- New attack pattern for normal enemies
- Adding new stages
- Adding life system for player
- Adding weapons system for player

5. System Model

5.1. Use Case Model



5.1.1. Play Game

Use Case Name: Play Game
Primary Actor: Player

Stakeholders and Interests:
-Player wants to play the game.
-System keeps the score of the Player.

Pre-condition: -
Post-condition: If player gets a higher score than any of the high scores system will update the high score list.

Entry Condition: Player selects "Play Game" button from main menu.
Exit Conditions: Player selects "Return to Main Menu" from Pause Menu. Finishing the game.

Success Scenario Event Flow:
1. Game is started by System.
2. Player starts playing from first level.
3. Player plays the stage until every enemy including the boss is defeated.
4. System grants access to next level.
5. Player starts playing next level.
6. If player gets a higher score than any of those in the high scores list system updates the high scores list.
7. System returns to main menu.

Alternative Flows:
- A. Player tries to destroy all enemies to complete the stage.
  - A.1. Player dodges the enemy projectiles with the keyboard.
  - A.2. Player shoots the enemy.
  - A.3. Enemies that collide with player's projectiles are removed if their health becomes 0.
  - A.4. *Player follows steps A.1 – A.3 to destroy all enemies.*
  - A.5. Player completes the stage, system updates player's score according to time.

- B. Player wants to pause the game anytime during the game.
  - B.1. Player presses Esc key to pause the game.
  - B.2. System pauses the game and shows "PAUSED" string.
  - B.3. Player wants to return to the game.
    - B.3.1. Player presses Esc key to resume the game.
    - B.3.2. System starts the countdown from 3.

5.1.2. Tutorial

Use Case Name: Tutorial
Primary Actor: Player

Stakeholders and Interests:
-Player wants to learn mechanics of the game
-System gives an introductory gameplay that helps the Player learn the mechanics

Pre-conditions: -
Post-condition: -

Entry Conditions: Player selects "Tutorial" from main menu. If it is first time the player tries to play the game
Exit Condition: Player selects "Return to Main Menu" from Pause Menu

Success Scenario Event Flow:
1. System shows the movement mechanics.
2. System shows the attack mechanics.
3. System Shows scoring system.
4. System shows power up system

Alternative Flows:
- A. If player desires to return main menu at any time.
  - A.1. Player selects "Return to Main Menu" from Pause Menu
  - A.2. System displays main menu.

5.1.3. Change Settings

Use Case Name: Change Settings
Primary Actor: Player

Stakeholders and Interests:
-Player wants to change game settings.
-System updates the settings which are changed by player.

Pre-condition: If Player changes game settings, adjusted settings will be saved and used by system.
Post-condition: Game settings are changed.

Entry Condition: Player selects "Change Settings" button from main menu.
Exit Condition: Player selects "Back" to return to main menu.

Success Scenario Event Flow:

1. Player presses "Change Settings" button to make changes about game settings.
2. Game settings are displayed to Player in "Change Settings" screen by System.
3. Player adjusts settings according to his desire.
4. System updates game settings successfully.

Alternative Flows:

A. If Player wants to return to main menu at any time.
   A.1. Player selects "Back" button from "Change Settings" screen.
   A.2. Game settings are updated by System.
   A.3. Player returns to main menu.

5.1.3.1. Change Player Name

Use Case Name: Change Player Name
Primary Actor: Player

Stakeholders and Interests:
-Player wants to change the player name.
-System updates the player names for all of the players.

Pre-condition: For first running, game settings will be set as default. If Player changes player name, new names will be saved and used by system.
Post-condition: Game settings are updated.

Entry Condition: Player selects "Change Settings" button from main menu.
Exit Condition: Player selects "Back" to return to main menu.

Success Scenario Event Flow:

1. Player presses the text field the corresponds to their name.
2. Player writes the new name.
3. Player clicks anywhere on screen.
4. System updates player's name successfully.

Alternative Flows:

A. If Player does not want to change the name.
   A.1. Player presses the escape key to restore the old name.

5.1.3.2. Change Key Bindings

Use Case Name: Change Key Bindings
Primary Actor: Player

Stakeholders and Interests:
-Player wants to change the control keys for any player.
-System updates the key bindings which are changed by player.

Pre-condition: For first running, key bindings will be set as default. If Player changes the key bindings for any player, newly selected keys will be saved and used by system.
Post-condition: Key bindings are updated.

Entry Condition: Player selects "Change Key Bindings" button from settings menu.
Exit Condition: Player selects "Back to Settings Menu" in the selection pop-up to return to settings menu.

Success Scenario Event Flow:

1. Player presses "Change Key Bindings" button to make changes about spaceship selection.
2. Key bindings menu is shown as a pop-up on top of the settings menu.
3. Player selects the key bindings for all of the players.
4. Player presses "Back to Settings Menu" to settings menu in selection menu.
5. System updates spaceship selections successfully.

Alternative Flows:

A. If Player wants to return to settings menu at any time:
   A.1. Player selects "Back to Settings Menu" button from the pop-up.
   A.2. Game settings are updated by System.
   A.3. Player returns to settings menu.

5.1.3.3. Change Spaceship

Use Case Name: Change Spaceship
Primary Actor: Player

Stakeholders and Interests:
-Player wants to change the spaceship for any player.
-System updates the spaceship selections which are changed by player.

Pre-condition: For first running, spaceship selection will be set as default. If Player changes the spaceship for any player, newly selected spaceships will be saved and used by system.
Post-condition: Spaceship selections are updated.

Entry Condition: Player selects "Change Spaceships" button from settings menu.
Exit Condition: Player selects "Back to Settings Menu" in the selection pop-up to return to settings menu.

Success Scenario Event Flow:

1. Player presses "Change Spaceship" button to make changes about spaceship selection.
2. Spaceship selection menu is shown as a pop-up on top of the settings menu.
3. Player selects the spaceships for all of the players.
4. Player presses "Back to Settings Menu" in selection pop-up.
5. System updates spaceship selections successfully.

Alternative Flows:

A. If Player wants to return to settings menu at any time:
   A.1. Player selects "Back to Settings Menu" button from the pop-up.
   A.2. Game settings are updated by System.
   A.3. Player returns to settings menu.

5.1.3.4. Change Volume

Use Case Name: Change Volume
Primary Actor: Player

Stakeholders and Interests:
-Player wants to change the volume of background music and sound effects.
-System updates the setting of volume as the user desires.

Pre-condition: For first running, volume setting will be set as default. If Player changes volume setting, adjusted volume will be saved and used by system.
Post-condition: Volume setting is updated.

Entry Condition: Player presses the volume slider in settings menu.
Exit Condition: Player releases the volume slider in settings menu.

Success Scenario Event Flow:

1. Player presses the volume slider in settings menu.
2. Player adjusts the volume setting by moving the slider.
3. Player releases the volume slider in settings menu.
4. System updates game settings successfully.

Alternative Flows:

A. If Player wants to does not change the volume setting in any time:
   A.1. Player moves the slider into it's old state.
   A.2. Player releases the volume slider.

5.1.4. High Score

Use Case Name: High Scores
Primary Actor: Player

Stakeholders and Interests:
-Player wants to see high scores
-System shows top 10 high scores

Pre-conditions: System keeps records of top ten scores.
Post-condition: -

Entry Condition: Player selects "High Scores" from main menu
Exit Condition: Player presses "Back to Main Menu" to return to main menu

Success Scenario Event Flow:

1. System shows top 10 high scores.
2. Player presses "Back to Main Menu" button.
3. System shows main menu.

Alternative Flows:

A. If player desires to return main menu at any time.
   A.1. Player selects "Back to Main Menu" button to return main menu.
   A.2. System displays main menu.

6. Conclusion

In this report, main focus was on the general properties of Astunc and its functions. But this report does not include the information about implementation of this parts. First main features of Astunc presented than above those use cases and scenarios were easy to construct. Game features and given requirements were our main helper before we started to consider use cases and their scenarios. In future this report will be useful in design process Those scenarios and use cases will led us to pass the coming design process easier and will help us the discussions about contents of the design report. In this analysis, as a case tool Visual Paradigm and as a modeling language Unified Modeling Language (UML) used. We used Visual Paradigm to create our UML diagrams such as Use Case diagrams.
To sum up about Astunc, this game will be played in such way that player will have to consider and dodge multiple enemies and gigantic number of projectiles from those enemies from the beginning to the end of the game. Ranking will be determined by the time instead of traditional "kill to gain point" type of scoring system. Which will add a uniqueness to the Astunc.