Dungeons and Dragons: Deliverable #1
SE 3A04: Software Design II – Large System Design
Group 6, Tutorial 2

Saad Ali - alis76

Aiyuan Liu - liua70

Harry Fu - fuh6

Francis Bajamunde - Bajamunf

Mohammed Mirajkar - Mirajkam

Shardool Patel - pates25

Contents

1 Introduction

1.1 Purpose

The purpose of the SRS document for Dungeons and Towers is to describe how the software of this tower defense game will be developed. This document will provide the necessary descriptions and requirements for the final implementation. Stakeholders and developers will have access to this document to understand the product-to-be and how it will be implemented.

1.2 Scope

The purpose of this product: Dungeons and Towers, is to entertain the user through unique and challenging gameplay. The game takes inspiration from Bloons Tower Defense and dungeon explorer games, both of which provide an engaging and unique twist to classic tower defense games. Our objective is to provide the user with an enjoyable and memorable gaming experience, along with great replayability. The game will access the users local memory to save game state and it shall run on Android Devices only.

1.3 Definitions, Acronyms, and Abbreviations

1. **Gold:** The resource that is used within a level to buy *Towers* and spells to use for that level. The user is assigned a different amount of starting gold each level. The user can also gain gold during the level by killing enemies with *Towers* and spells.
2. **Levels:** The setting in which gameplay primarily takes place. It is where *Towers* are placed, enemies move, and where spells are activated. Divided by different difficulties. Each level has an individual difficulty, with higher levels being more difficult. The user can unlock the levels manually by playing through them.
3. **Towers:** Objects that can be placed that shoot projectile objects. These projectiles will cause damage that reduces the total health-points of incoming enemies. Different towers have different base stats and effects such as projectile speed, projectile damage, and health-points.
4. **Deck:** Subsets of unlocked towers and spells that the user has selected prior to starting a level. Users will have two separate decks; one for *Towers* and one for *Spells.*
5. **Spells:** Includes different abilities that change gameplay such as freezing, burning, or slowing down enemies. They can apply to the towers and cause damage to enemies.
6. **Wave:** Each level has a defined number of waves. Each wave has a different amount of enemies. The types of enemies in each wave vary.
7. **Enemies:** In-game objects that serve as the player's obstacle to victory. Enemies are varied and can have different properties such as different movement speed, attack-damage, or health-points. Defeating all enemies in a level will result in victory in that level.
8. **Health-points (HP):** An integer attribute given to in-game objects such as enemies and towers. An object is eliminated from the level when their total-health points reach zero.
9. **Timer:** In-game tool that counts the time since each wave started. The timer will be displayed at the top of the gaming interface.
10. **Unity:** Cross-platform game engine developed by Unity Technologies that can be used to create Android games.
11. **Area of Defense:** Specified range around a given tower. When enemies enter a tower's area of defense, the tower will attack that enemy until it is either eliminated or it exits the area of defence.

12. **Area of Effect:** Specified range where a spell will execute its specified effects. When an enemy object enters a spell's area of effect, the enemy will experience whatever effects the spell is specified to have. The spell will continue to affect the enemy until it is either eliminated or exits the spell's area of effect.

13. **Map:** A combination of all levels, contains all the paths enemies travel through in the game along with areas to place towers.

14. **Result of the level:** The outcome of the level. Successful if the user completes all waves without getting to 0 HP. Unsuccessful if the user, at any time during the level, gets to 0 HP.

15. **Placed:** Refers to the towers and spells being positioned on the map.

16. **Selectable:** In-game objects that the user can interact with. The user has the ability to purchase, play or modify the objects by clicking on it.

1.4 References

1. SE3A04 tutorial slides2 and slides3 by Andrew LeClair

1.5 Overview

The remaining sections of this document provide a general description, including characteristics of the user of this product, the products description, as well as the functional and nonfunctional requirements. General description of this project is discussed in section 2 of this document. Section 3 gives the business events and functional requirements while designing the game. It also specifies the viewpoint of the system. For section 4, it describes all the nonfunctional requirements along with more supporting information.

2 Overall Description

2.1 Product Perspective

Similar to this products inspiration, Bloons Tower Defense, this product will remain a self-contained mobile application with no dependencies towards external applications or resources.

2.2 Product Functions

The primary game piece are the deck slots, one consisting of towers, and one consisting of spells. A level is the core gameplay where the user must survive the incoming wave using their respective *Towers* and spells in their decks. After the end of each level, the user will be prompted to choose the next level from a variety of choices. The main menu will function as the home screen. It will allow the user to modify game settings and initiate gameplay.

A unique and innovative feature of Dungeons and Towers is its replayability. Every time a user plays this game, the *Towers* unlocked and the spells acquired will be different as the user is presented with a random selection at the end of each level. This means that the user will have to adjust their strategies to beat the same level they have beaten before. Additionally, upon finishing the main game, there will remain unexplored locations along unchosen paths to the end which the user has not played before. Overall, this feature enables users to have unique experiences with every fresh start of the game.

2.3 User Characteristics

The user is expected to be familiar with the hardware on which the application will be available. They shall also have the physical and visual ability to interact with the application and to perform the actions the application specifies.

2.4 Constraints

This game is implemented in *Unity* and will function on an Android mobile device. Access to the Google Play Store is required to install this game and subsequent updates/bugfixes. This game supports one player per device.

2.5 Assumptions and Dependencies

(i) *Unity* continues to support development and deployment to the Android Mobile platform.

(ii) User has updated to an Android version that *Unity* is currently supporting.

2.6 Apportioning of Requirements

This system shall be available only in English.

3 Functional Requirements

VP1. Viewpoint: User

BE1. **The user wants to start the game**

- FR.1 The application must present an option for the user to start a new game.
- FR.2 The application must present options for the user to continue from the last save state.
- FR.3 If no previous save state exists, a new game shall be started for the user.

BE2. **The user wants to modify options**

- FR.4 The user must be given the option to access the game settings at all times in the main menu and during *gameplay*.
- FR.5 The game must be paused when the user accesses the game settings during *gameplay*.
- FR.6 The user must be allowed to change the games audio levels and contrast.

BE3. **The level map is loaded**

- FR.7 The game must give the user the option to change their deck before the level begins.
- FR.8 The application must load the level and start the first wave upon user confirmation of their decks.
- FR.9 The game must present the user a view of their decks throughout the level.
- FR.10 The level map is displayed.
- FR.11 The game must present the user their *starting level conditions*.
- FR.12 The game shall allow user to place towers and use one-time spells if the gold requirements are met.
- FR.13 The user must be allowed to initiate the level *gameplay*.

BE4. **The level gameplay begins**

- FR.13 The level must end when an enemy reaches the end of the *map*.
- FR.14 The level must end when the whole wave is defeated.
- FR.15 The enemies must follow the *map* path .
- FR.16 The elimination of an enemy object must increase the player's gold by an amount dependent on the object that was eliminated.
- FR.17 The game shall allow the user to pause the game at any point.

FR.18 The towers must shoot projectiles at enemies within their area of defense.

FR.19 When an enemy is hit by a projectile they must lose health points.

FR.20 When an enemy's health points reaches zero the enemy must be eliminated.

BE5. **The level gameplay has ended.**

FR.21 The *result of the level* must be presented to the user.

FR.22 If the level was successfully cleared, the user must be presented with new towers and spells to pick, along with a choice of at most 3 possible paths to different levels.

FR.23 The game shall unlock the towers and spells that the user selects as their reward. These towers and spells shall be available for the user to add to their deck.

FR.24 Half of the levels in the game must be *muti-path levels.*

FR.25 If the user failed to complete the level, they must be given an option to restart the current level.

FR.26 Half of the levels in the game must be *muti-path levels.*

FR.27 Once a choice is made, the new level must be *loaded.*

BE6. **User wants to use a *Towers* during a level**

FR.28 Unlocked Towers must be *selectable.*

FR.29 A selected Tower must present its stats and available upgrades.

FR.30 A selected Tower must present its area of defense.

FR.31 Selected towers must only be able to be placed on valid areas of the map.

FR.32 The game must inform the user whether desired placement of a tower on the map is a valid placement.

FR.33 Selected Tower must allow the user to confirm the placement or cancel the selection.

FR.34 Towers must not be able to be moved once they are *placed.*

FR.35 The game shall allow the user to sell placed towers.

BE7. **User wants to use a spell during a level**

FR.36 Spell cards must be *selectable.*

FR.37 A selected spell must display its stats and description of its effects.

FR.38 A selected spell must display its *area of effect*

FR.39 A selected spell must be able to be *placed* on valid areas of the map.

FR.40 A *placed* spell must activate its effect immediately.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

LF1. The levels and objects shall be two-dimensional in design.

LF2. The map shall be minimalistic in design.

LF3. The level design must utilize colors and props that present a medieval setting.

LF4. The *tower deck* and the *spell deck* must be nicely separated on the user interface.

LF5. The towers shall have distinguishable shapes to show their identities.

LF6. The game instructions and given button shall be easy to understand and approachable.

4.1.2 Style Requirements

LF1. The audience must interpret the design to be medieval in setting and tone, while the gameplay itself provides a simple feel that allows for more casual play.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

UH1. There must be a help menu that list the description and effects of the towers and spells.

UH2. The game state must be saved automatically upon the end of each level.

UH3. The game shall follow all the user interface rules of existing tower defence games.

4.2.2 Personalization and Internationalization Requirements

UH5. Users must be able to adjust the volume of background music and the contrast of the application.

4.2.3 Learning Requirements

UH6. The game must provide a tutorial for the players in order to help them get acquainted with certain game mechanics.

4.2.4 Understandability and Politeness Requirements

UH7. The product shall use the symbols and words which are understandable by the user community.

4.2.5 Accessibility Requirements

UH8. The game shall be accessible for partially sighted users, deaf users and red-green colorblind users.

UH9. This game shall be playable by individuals with at least one functioning hand.

UH10. Users must be able to operate an android running phone with a touchscreen.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

PR1. Any valid user input shall receive an immediate response by the game in a manner instantaneous to the user's perception (under 0.5 seconds).

PR2. Game initialization upon opening the application shall be less than 10 seconds.

4.3.2 Safety-Critical Requirements

Not Available

4.3.3 Precision or Accuracy Requirements

PR3. The health points of object and enemies must be an integer value.

PR4. The damage number must be an integer value.

PR5. The gold must be an integer value.

4.3.4 Reliability and Availability Requirements

PR4. The software must be able to initialize and operate normally, not necessarily continuously, at any time.

PR5. This game shall be available to play whenever the user intends as there is no requirement for an internet connection.

PR6. Internet connection must be present when a user wishes to report feedback.

4.3.5 Robustness or Fault-Tolerance Requirements

PR6. In the event of a crash, upon reboot, the last saved game state shall be loaded.

PR7. In the event of a corrupt save file, the user must restart from the beginning.

4.3.6 Capacity Requirements

PR8. The product shall be designed primarily for single player only.

4.3.7 Scalability or Extensibility Requirements

PR9. The storage capacity of the product shall increase to accommodate future updates and potential bug-fixes.

4.3.8 Longevity Requirements

Not Available.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

OE1. The product is shall function on Android running mobile devices.

OE2. Dungeons and Towers must be played in landscape position on mobile devices.

4.4.2 Requirements for Interfacing with Adjacent Systems

Not Available

4.4.3 Productization Requirements

OE3. Upon completion, the product shall be released on Google Play Store for Android devices.

4.4.4 Release Requirements

OE4. Updates and maintenance releases must occur whenever bug-fixes or updates are made to the games software

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

MS1. The properties of game elements shall be updated appropriately to ensure game balance.

4.5.2 Supportability Requirements

MS2. An FAQ will be available for public use to answer commonly asked questions.

MS3. A tutorial will be made available at the beginning of the game in order to help the user get accustomed to gameplay mechanics.

MS4. Bug reports must be implemented to allow users to report bugs and glitches that need fixing.

4.5.3 Adaptability Requirements

MS3. In the future it shall be implemented in multiple platforms and will keep all the existing features.

4.6 Security Requirements

4.6.1 Access Requirements

SR1. Users shall be restricted from access to certain information such as feedback and game data.

4.6.2 Integrity Requirements

SR2. Users must not be able to modify data of local files.

4.6.3 Privacy Requirements

SR3. The product shall not interact with any personal user data and just modify the game local data on their cell phones.

4.6.4 Audit Requirements

Not available

4.6.5 Immunity Requirements

Not Available

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

CP1. The product shall not contain any religious imagery or text.

CP2. The product shall not contain any references to any national disaster.

4.7.2 Political Requirements

Not available

4.8 Legal Requirements

4.8.1 Compliance Requirements

LR1. This software shall comply with all national and federal software regulation laws

LR2. This software shall comply with all relevant software standards

LR3. This software shall comply with all relevant privacy acts.

4.8.2 Standards Requirements

LR4. The product shall comply with basic programming standards.

A Division of Labour

Saad Ali - Non Functional requirements, Introduction, General Contributions.
Aiyuan Liu - Functional requirements, General Contributions.
Harry Fu - Non Functional requirements, Overall Description, General Contributions.
Francis Bajamunde - Non Functional requirements, Introduction, General Contributions.
Mohammed Mirajkar - Functional requirements, Product Perspective, Product Functionality, Business Events, General Contributions.
Shardool Patel - Functional requirements, Business Events, Introduction, General Contributions. [illegible]