**Requirements and Analysis Document for NNN**  
Table of Contents  
  
 Version:   
  
 Date   
  
 Author   
  
This version overrides all previous versions.  
  
**1 Introduction**  
  
This section gives a brief overview of the project.

**1.1 Purpose of application**

The purpose of this project is to create a tower defence desktop application. Our primary goal is to create a very basic version of tower defence, and if time or difficulty are not obstructing our workspace implement more advanced features.

**1.2 General characteristics of application**

The project goal is to develop a 2D Tower Defence game - a type of game in which the player tries to kill all enemies moving along a given path. If an enemy reaches the end of the path without being killed, the player loses lives. The player places various towers on the game map, and has the possibility to upgrade or sell them if he/she has enough money, which is earned by killing enemies. The game is over when all the player’s lives are lost.

**1.3 Scope of application**

The progress of the game will not be saved if one decides to quit the program during a session. Due to the nature of the tower defence, there will only be a single player mode.

**1.4 Objectives and success criteria of the project**

The success criteria is to have a playable game at the deadline of the project. This includes the ability to play at least five waves of enemy attacks in a row. Also, having at least two different kinds towers is necessary. One or more of these towers has an effect associated with it, which may be applied to the enemy it shoots. At least two different kinds of enemies are required. **1.5 Definitions, acronyms and abbreviations**

Enemy - Thing that needs to be killed before it reaches the end of the path.

Player - The person that plays the game.

Path - The path that the enemies follow across the map.

Map - A top down representation of a map. Contains a path and a build area.

Game over - The player’s game session ends if the player has no more lives. Prompts the user to play the same map again or to select a new map.

Tower- An automated sentry-like object which shoots and damages enemies. Is bought by the player and can be upgraded by the same person, these actions requires money. Can only be placed in the build area of the map.

Money- Money is used to upgrade or buy towers. The player gets money from defeating enemies.

Status effect - an effect that changes the status of an enemy, e.g. decrease the speed of the enemy. **2 Requirements**  
In this section we specify all requirements  
  
**2.1 Functional requirements**  
Create a list of high level functions here (from the use cases).  
  
**2.2 Non-functional requirements**   
  
Possible NA (not applicable).  
  
**2.2.1 Usability   
  
2.2.2 Reliability   
  
2.2.3 Performance   
  
2.2.4 Supportability  
  
2.2.5 Implementation   
  
2.2.6 Packaging and installation  
  
2.2.7 Legal   
  
2.3 Application models  
  
2.3.1 Use case model**  
UML and a list of UC names (text for all in appendix)  
  
**2.3.2 Use cases priority**  
A list  
  
**2.3.3 Domain model**  
UML, possible some text.  
  
**2.3.4 User interface**  
Text to motivate a picture.  
  
**2.4 References**  
 APPENDIX   
  
 GUI  
  
 Domain model  
  
 Use case texts