Project Analysis

# Project Management

* List activities up to this point with the time invested for it
* Compare to time spent to the time planned
* List measures to compensate if too much time was spent on something
* Show detailed planning for the next milestone
* Show risk list again (with updates if necessary)

# Use Cases

* List all use cases with priorities
* Show most important use case “fully dressed”
* All other important ones shall be “casually dressed”
* The rest can be “brief”

Use Case UC1: Play game

**Primary Actor:** Player

**Stakeholders and Interests:**

* Player: Wants a stable framerate with short load times and no crashes to interrupt his experience.

**Preconditions:** The player has selected a map and started a new game on it.

**Post conditions:** The player has either defeated all enemy waves and won or his central structure has taken a critical amount of damage and has been destroyed resulting in the player losing the game.

**Main success scenario:**

1. The player has started a new game and the map is loaded.
2. The player spends his starting currency on building up his defences.
3. The player clicks on begin, indicating they are done preparing and ready for the first enemy wave.
4. Incoming enemies are destroyed by the defensive structures and the player spends the money gained on new defences.

*Step four repeats itself so long as there are enemies remaining in the current wave and the main structure has not been destroyed.*

1. When all enemies of the current wave have been destroyed, there is an indication that the next wave will be incoming soon.
2. The player has a set amount of time to improve his fortifications before the next wave begins automatically.

*Steps four through six are repeated while the last wave has not been defeated and the main structure has not been destroyed.*

1. The player has defeated the last wave and a message is displayed indicating that they have won the game.
2. The game automatically returns to the main menu after the message has disappeared.

**Extensions:**

\*a. At any time, the game crashes:

The game shuts down and the player must restart the game if he wishes to continue playing. Any game progress will not be saved.

\*b. The player closes the game window:

The current game ends and no game progress is saved.

\*c. The Player pauses the game:

The ongoing game is paused and a menu is brought up allowing the player to leave  
 to leave the game or to resume it.

**Special Requirements:**

* Windows or Mac computer with Java 8.
* Computer with mouse or a touch display.

**Frequency of Occurrence:** However often it is initiated by player.

# Use Case Diagram



# Domain Model

* A domain model that shows the most important concepts from the use cases
* A detailed description of the Problemdomäne <https://de.wikipedia.org/wiki/Problemdom%C3%A4ne> that is derived from the domain model

# First Architecture

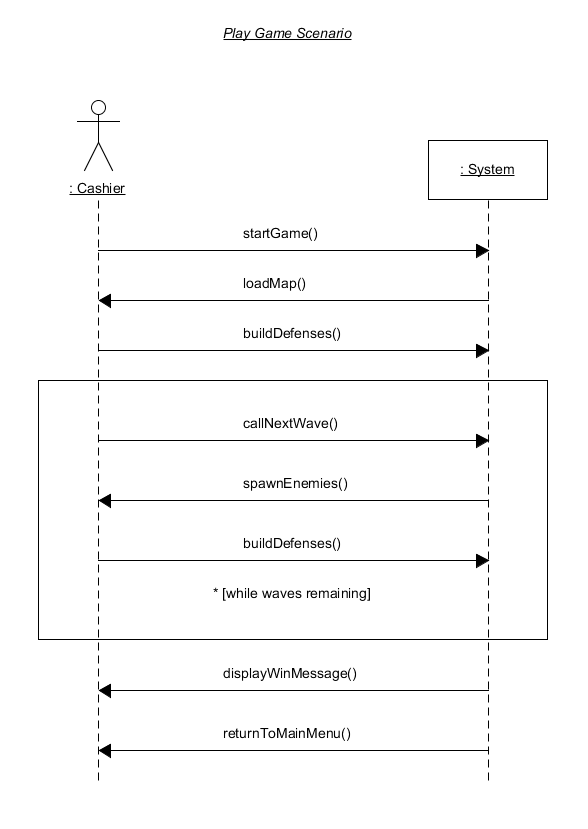
* The first architecture should explain the project idea (Standalone-App, mobile App, with or without server, P2P, 2D or 3D GUI, etc.)
* This should cover the most important use cases

# Additional Specifications

* List additional functional and mainly non-functional requirements according to the FURPS+ model
* List important game rules
* List other important information (for example conditions related to development, etc.)

# System Sequence Diagram

* The fully dressed use case must be formulated as a sequence diagram

[ 🡨 WORK IN PROGRESS ]

# Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Castle | The players central structure, the defense of which is the games main objective. |
| Tower | Any of a variety of defensive or offensive building created by the player to hinder or destroy incoming enemies. |
| Wave | A wave refers to a group of enemies. A game encompasses multiple waves that need to be defeated. |
|  |  |
|  |  |
|  |  |
|  |  |