Project Plan, Group 38

Infected Zoo

Version 1.0

2015-03-19

**Table of Contents**

1. Product description
2. Project Overview
3. Crew
4. Project activities
5. Risk analysis
6. Deliveries

**Purpose of the document**

The purpose of this document is to describe the finished product of this project, the plan we (the developers) intend to follow. The document will also show the timeframe, graphically shown in a Gantt-chart, and deliveries of the project and it’s individual components.

**Scope of the document**

The document contains a rough plan for the project and it’s stages but it does not contain a detailed plan for each stage in the process.

The document does not contain a requirement specification, those requirements that do exist does so in a separate document.

In this document exist a brief description of the finished product. The detailed design of it does however not.

**Product description**

We are creating a web based platformer game.

The web site contains a homepage where the game is located and can be played directly in the browser, a comments page(chat), an about the project page and a page with leaderboards.

The game is a 2D-platformer, where friends and foes alike have a battle to the death, with cute animals infected with horrible viruses. Each match will generate statistics that will be used on the leaderboards-page on the web.

**Project overview**

**Purpose of the project**

Get a passed grade in the course and learn python, Flask and more about web development. Learn how to use the Unity-game engine. Learn the basics in working in bigger projects.

**Scope of the project**

Players will be able to play a fun game with their friend or nemesis to pass the time.

On the website the user will get the opportunity to read more about the game and the characters in it. Statistics from each match can be saved and displayed on the website with the name the user chooses, similar to arcade machines.

It will not be possible to download the game from the website or to have a profile for statistics.

**Objectives of the project**

To make a game for at least two players playing from the same keyboard. The game should be available on a website which shall contain features and information related to the game .

If there is more time we plan to implement facebook features like an online multiplayer mode, log in and creating a profile, and sharing stats .

**Target audience**

Our target audience is people who wants to play a simple and enjoyable game for an hour or two, simply to kill some time.

# **Crew**

The team will consist of two information architects, Malin and Lisen, who will create the website, and two game developers, Joakim and Alexander, who will create the game.

**Project** **activities**

**Website**

1. flowchart 5h x 2p v.12
2. wireframe 7h x 2p v.13
3. class diagram 5h x 2p v.13
4. HTML prototype 10h x 2p v.14
5. Python 30h x 2 v.14-15
   1. making the web page functional

**MILESTONE** - Functional web

1. comments page 20h x1p v.16
2. leaderboard 20h x 1p v.16
3. CSS 10h x 2p v.17
4. responsive design 10h x 2 v.17
5. database implementation 20h x 2 v.18

**DEADLINE !!!**

1. Optional
   1. Facebook features til the end

**Game**

1. single player prototype 4h x 2p v.12
2. art 40h x 1p v.13-14
3. local multiplayer 20h x 1p v.13-14

**MILESTONE** - HTML-product

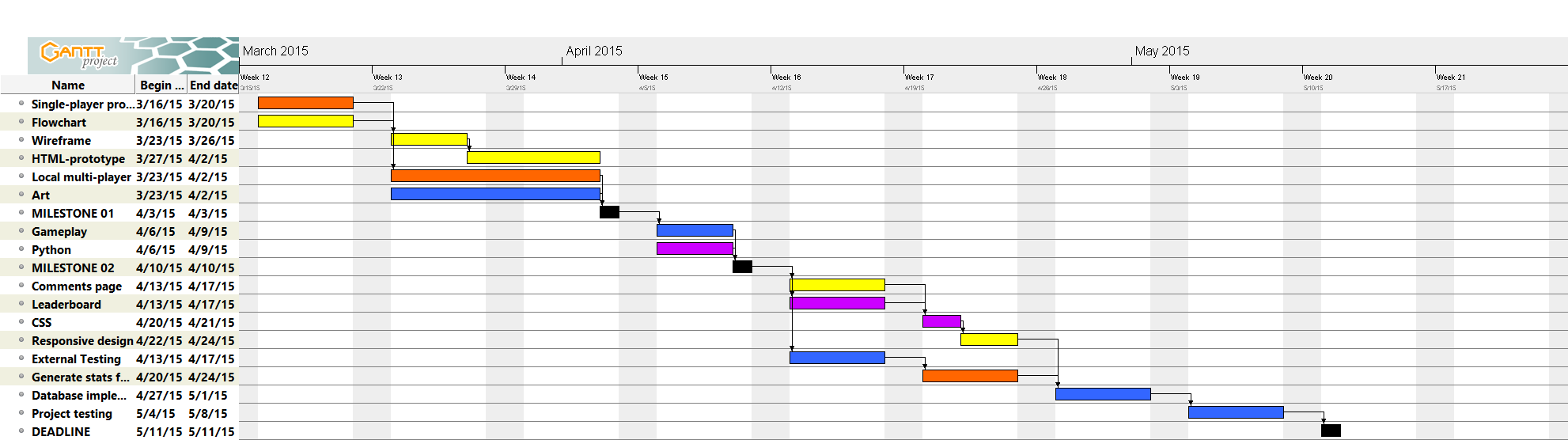
1. Gameplay 20h x 2p v.15

**MILESTONE** - Game done

1. external test 20h x 2p v.16
2. generate stats for leaderboard 10h x 2p v.17
3. database implementation 20h x 2p v.18

**DEADLINE !!!**

1. Optional
   1. Facebook features til the end



**Responsibilities**

**Malin** (yellow) - Wireframe, HTML prototype, Comments page, CSS, Facebook, logo

**Lisen** (purple) - Class diagram, CSS, Python, leaderboards page,

**Alexander** (orange) - game prototype, local multiplayer, generate stats

**Joakim** (blue) - Art, gameplay, external tests, database implementation, Facebook

# **Risk analysis**

**Risk:** Learning new Software (Unity/ Flask) is an early risk factor as it requires the team to work with programs that they lack experience with.

**Countermeasure:** a combination of teacher guidance and self tuition with online guides/youtube guides to learn how to interact with the new software.

**Risk:** Inexperience with new code,working methods, and new things in general.

**Countermeasure:** Working together with the team to overcome new things and gain insight as well as knowledge in new fields of study.

**Risk:**.Technical difficulties, such as hardware failure, software failure, loss of files etc.

**Countermeasure:** Store all files and backups on multiple computers, or on a cloud server that every member of the team have access to.

**Risk:** A team member get ill. While this isn’t linked to the project itself, it still impacts it as a whole thus making it a relevant risk.

**Countermeasure:** There isn’t much to be done with illness within the team other than recuperate with plenty of bed rest etc. Other team members might have to be prepared for heavier work load. Alternatively the sick team member can do light work tasks from home like documentation.

**Risk:** Disagreement amongst the team members might be a serious risk as it can complicate the teams ability to work together as a cohesive unit.

**Countermeasure:** Work out compromises amongst the team, find common ground and work out a cohesive plan for the project, make sure to have meetings regularly as the project progresses.

**risk:** the comments page will be hard to do when we plan to it because we won’t know enough about databases at that point.  
**Countermeasure:** do it later.

**Deliveries**

**web**

**Flowchart:** describing what’s on every page och the web plattform.

**Wireframe**: high fidelity wireframe with theme, interface design, navigation design, information design.

**Class diagram** - defining and structuring ID’s and classes for the CSS. a map or document to follow when we’re doing the CSS.

**HTML:** all the HTML templates done.

**Python:** all the HTML templates put together and working.

**WEB MILESTONE 1:** working website without CSS.

**Comments page:** a comments page where the user writes a comment with a name. when the user publish the comment the comments flow shall update immediately.

**Leaderboard**: a page where the user writes a name to the new highscore and then the leaderboard updates with the 10 best results.

**CSS**: making the page look good, be user-friendly and easy to navigate.

responsive: the game window shall be centered at all types and sizes of screens. The webpage should not look too bad if you open it och a tablet or a phone.

**Game**

**Singleplayer prototype:** Create a simple 2D-platformer in Unity, where a character is controlled with the keyboard, collides with platforms and can fall to its death.

**Art:** Draw sprites and background images for use in the game.

**Local multiplayer:** Expand the single-player prototype by adding a second character, creating a killing mechanic where one player can kill the other by jumping on its head, adding respawn-points across the playfield and randomizing the respawn location each death. Also add a start screen and an options menu.

**GAME MILESTONE 1:** Build a multiplayer game with the art implemented for the website to use and implement.

**Gameplay:** Finish the last things on the game. Adding winning conditions, HUD.

**GAME MILESTONE 2:** Finished game, ready for testing.

**External test:** Find bugs, things to improve gameplay-wise and check if all the requirements for the game are met.

**Generate stats for the web:** Take stats from the game and create an external file for the website to use and post.

**Document history**