Cm222qk Christoffer Malmberg

# Project plan

**Vision:**

Creating an application of a game of hangman. The application will support a set of words that will be loaded in from a text file provided with the project. Words from that text file will be randomly chosen when starting a new game. The application will have graphical output showing how many letters the word contains, what their previous guess has been and how close players are to getting ”hung”.

The application will also support multiplayer in a way that where players can input a word of their choice or randomize a word from the preset list. The other player will then try to guess the word. If the player gets the word right he gets one point and the first player to get 3 points win.

The application will also have a menu that should be easy to navigated for the user.

**Introduction:**

A simple game of hangman containing key features for the game, multiplayer and an easily navigated menu.

**Justification:**

This application should be made because it fits in with todays standard on how much time people have to enjoy things, people today are very busy there for a quick game of hangman is perfect for busy people that just wants to relax for a bit and get enjoy something different.

There is also not really many well known hangman applications out there, and this project might just fill that gap.

**Stakeholder:**

The main stakeholders for this project are the tutors.

**Resources:**

The resources for the projects:

One programmer with assistance from teachers if needed.

About four weeks of dedicated time for the project.

Java coding using the ide Eclipse

**Hard and software req:**

The application will be developed using Java with the ide Eclipse, also some sort of graphics program like photoshop or paint will be used to get graphics for the application in the later stages of the development. Everyone should be able to run the application with pretty much any computer as it wont be a heavy application.

**Overall Project schedule:**

The project has four important deadlines:

1. Have some basic features implemented (”Skeleton” for the game) to start off with.
2. A lot of the main features implemented such as being able to play a full game, and some graphical output.
3. Finishing up the main features, testing it out so that it works well, add optional ”cool” features.
4. Fully finished well working application.

**Scope, Con and assumptions:**

Scope – To deliver a fully working application of hangman with all the core features need to play the game while also making sure every deadlines is met so that the application can be realsed in time.

Con – Basic features most be implemeted, make sure it is before implementing any other addition features! A half working game is not accaptable.

Assumptions – A fully working game with all it’s basic features, maybe some better graphical update torwards the end.

**Risk analysis:**

Risks:

1. Not dedicating enough time to certain features
2. Putting to much time into features that are not ”important” for the main game.
3. Making old features not work well when implementing new features.

Solution:

1. Not under estimating the important features! Make sure there is a lot of time for them.
2. Cut this feature if it’s not necessary! Put time into other features instead.
3. Remove the feature and try to rethink the implemention, don’t get stuck on one way there is multiple ways to implement certain features.

**Timelog:**

Implent skeleton code: est 2 hours