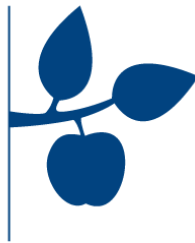


SEMESTER PROJECT IN COMPONENT-BASED SOFTWARE DEVELOPMENT

PROJECT FOUNDATION



UNIVERSITY OF SOUTHERN DENMARK

Casper Beese Nielsen
Exam № 409468
casni14@student.sdu.dk

Oliver Vestergaard
Exam № 64754140
olves13@student.sdu.dk

Jakob Tøttrup
Exam № 334604
jatoe13@student.sdu.dk

Simon Hjortshøj Larsen
Exam № 64754151
sila114@student.sdu.dk

Jesper Bager Rasmussen
Exam № 64754145
jrasm14@student.sdu.dk

Thomas Lemqvist
Exam № 410261
thlem14@student.sdu.dk

Project Supervisor: Jan Corfixen

2/23/2016



Table of Contents

Preface	I
Table of Contents	I
Main	1
1 Project-Y! Description	1
2 Existing Solutions	2
3 Learning Objectives	3
4 Tools & Methods	4
5 Time Schedule	4
6 Preliminary Report Outline	4
Appendices	A
7 Cooperation Agreement	A
8 Supervisor Agreement	B
List of Figures	C



1 Project-Y! Description

This section describes the product, which will be developed in this project.

The idea is to develop a top-down 2D shooter RPG game that contains, in its most basic form, a map where the player entity can move around and shoot enemy entities. The map can contain various obstacles, like impassable forests, roads where the player moves faster/further.

The idea is that the different entities can be developed as separate implementations. It makes the most sense to implement AI in the enemy component, which is where we will implement it.

We would like to start off with implementing the core of the game, and then develop additional implementations of the components (see Stretch goals).

The enemy and weapon components will be able to be interchanged with different implementations at runtime.

One way of implementing the map would be to use 2 dimensional arrays, with values determining what type of terrain a given pixel/tile is. For example an impassable forest could have value 5 and the player entity cannot pass through a value greater than 4 (could be made possible with "power-ups").

The game will be a "survival-mode", where the enemies spawn indefinitely and progressively get stronger and/or increase in numbers until the player is overwhelmed and dies. This death can be put off with stronger implementations of the player weapon/health as x amount of enemies are destroyed.

Features:

1. Procedurally Generated Maps
2. AI Enemies
3. Player and Enemy Weapons
4. Collision Detection

Stretch goals:

1. Different Weapons
2. Multiplayer
 - (a) Local
 - (b) Networked
3. Joystick support
4. RPG-elements
 - (a) Quests / Campaign
 - (b) Inventory
 - (c) Open world
 - (d) Levels

5. Local storage / Online storage

2 Existing Solutions

This section describes some already existing games, which has been used for inspiration.

Figure 1: A picture showing the differences in the same map from Diablo III



The diablo game world is randomly generated on runtime, which means that the map is never the same (see Figure 1), Project-Y! will incorporate a feature like the random map from Diablo, where the map is randomized as the player moves around in the game world.

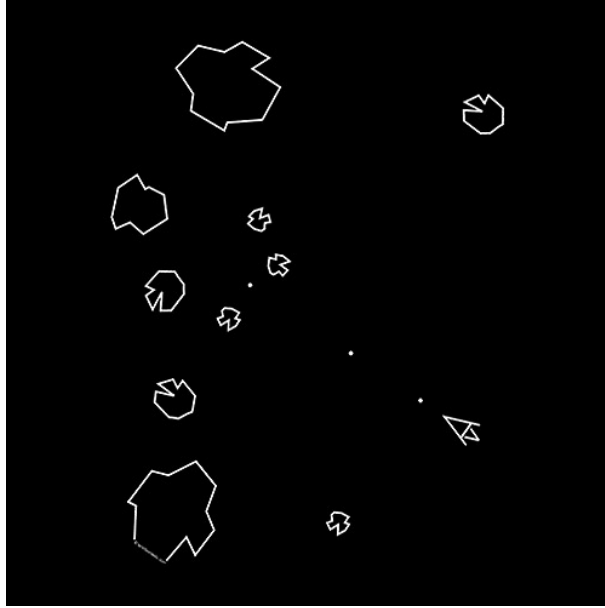
Figure 2: A picture showing the viewing angle in GTA 2



GTA 2 has a top down camera, where everything is seen from above (see Figure 2). ProjectY! will also have a top down viewing angle, so that everything is seen from above.



Figure 3: A picture showing the the stationary camera from Asteroids



Asteroids has a stationary camera, where the camera is standing still, and all the elements are moving around, this means that the camera is not following the player, and the player can move freely. In Project-Y! the camera shall be stationary, and when the player moves out of the camera sight, a new map is generated which the player enters from the appropriate side.

3 Learning Objectives

This section describes some learning objectives, that the group wishes to fulfill in this project.

KOM

- How can a component framework be used, to load and unload components during runtime in the game.

AI

- What kind of Artificial Intelligence (AI) would be optimal for this kind of game.
- How can an AI be implemented to play the game as an enemy.

DM507

- How can a random map be generated each time the player moves outside the map.



4 Tools & Methods

This section outlines some tools and methods, which will be used in the project. Suggested tools for developing the project

- JMonkeyEngine (JME) or LibGDX.
- Java OSGi.
- Github for VCS.
- Trello for SCRUM.
- Google Drive (docs) & L^AT_EX for report.

Suggested methods for developing the project

- SCRUM.
- UML for relevant diagrams.

5 Time Schedule

This section shows an early estimate of the project time schedule.

Figure 4: A gantt table showing the estimated time shcedule of the project.

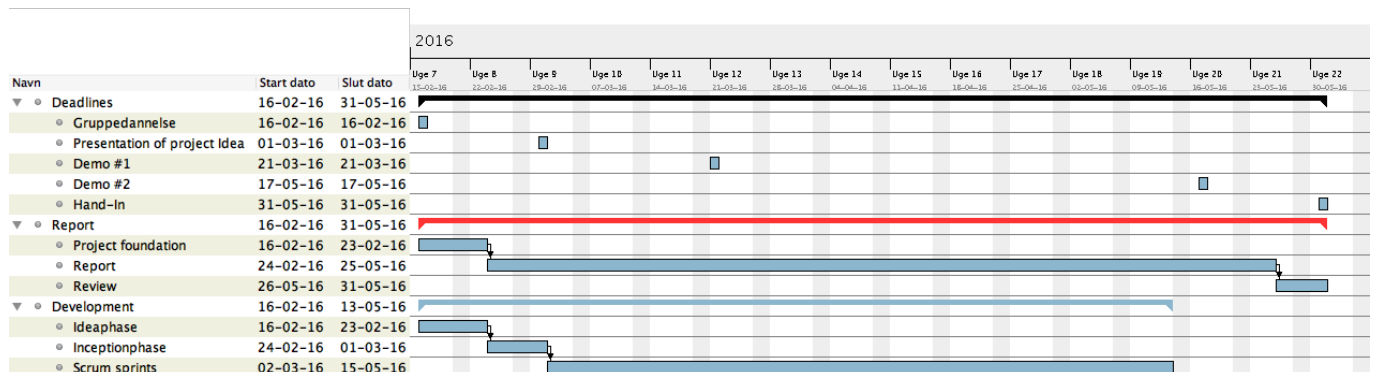


Figure 4 shows the preliminary time schedule for the project, it is divided into three parts: Deadlines, Report and Development.

6 Preliminary Report Outline

This section outlines the report.



1. Preface
 - (a) Abstract
 - (b) Editorial
2. Introduction
3. Problems and Goals
4. Tools
5. Analysis
6. Design
7. Implementation
8. Testing
9. Evaluation
10. Conclusion
11. Appendices
 - (a) References
 - (b) Appendices

7 Cooperation Agreement

This section contains the Cooperation agreement.

General:

1. Everyone should be on time. Rather 5 minutes too early than 5 minutes late.
2. Project work days are usually Fridays, but if necessary more days will be scheduled.
3. If a group member is running late they are required to inform another group member either by Facebook or mobile.
4. If a group member cannot attend the scheduled project workday, they are required to give notice asap. Every third time a group member is more than 5 minutes late, they must bring cake for the group, the next project workday.

Preparation:

1. The work material to be prepared before a project workday is agreed upon on the previous project workday. (In special cases, Facebook can be used to coordinate work material).
2. Group members are required to complete any task given to them, by the agreed upon deadline.
3. Group members needing help with their task should make sure to seek help from other members as soon as possible.



Agreement:

1. All decisions are decided by simple majority.
2. If the group members cannot find agreement, they will seek the help of their adviser.
3. The project will be written partly in Google docs and L^AT_EX.

Teambuilding:

1. The work pace should not be grueling, as there should be time to have fun and team build. That said, the work should be completed within the deadlines.
2. If a group member is working and you're taking a break, it is expected that you don't do anything to break their concentration. (Loud off topic discussions, music etc.)
3. Breaks can be coordinated within the group, or short breaks can be taken individually.

Level of ambition:

1. It is expected that every group member's ambition level should be high.
2. The ambition level can under special circumstances (deadlines, etc.) be changed, as long as the group agrees.

Learning:

1. It is up to the individual group member to take responsibility for his/her own learning.

Work Culture:

1. Group members should be respectful of each other. (Let them speak and listen to their points of view)
2. If a problem should occur between group members (academically or personally), it should be addressed as soon as possible.
3. Work problems should be solved by groups of two where possible.
4. The work tasks of the day should be clear and thought out beforehand.
5. At the end of the project workday, the results should be evaluated and the work tasks for next time should be planned.
6. In case of serious violation of the cooperation agreements, the advisor will be contacted.
7. It is expected that every group member attends lectures and actively seeks to further their understanding of Software Engineering, to help the group succeed.

8 Supervisor Agreement

This section contains a draft Supervisor agreement

The group emphasizes that it is the group itself, not the supervisor, who is responsible for the group's learning. However, the supervisor has a responsibility to push the group in the right



direction, as the supervisor can be consulted with doubts and questions. The supervisor can also provide feedback on written work the group has produced.

All members of the group are expected to participate in the meetings. Both the group and the advisor are expected to arrive prepared at meetings. At meetings, the group has prepared an agenda and the supervisor has read and addressed the possible issues submitted by the group. Emails from both parties are generally expected to be answered within one-two business days.

The group expects that there will be a meeting every week, unless there is no need for this. Supervisor meetings last 45-60 minutes and will take place on Tuesdays (3-4 pm). Meetings and agendas should be announced no later than the day before the instructions. The group should produce a transcript of the discussed topics after each meeting, which should then be distributed to group members and the supervisor.

List of Figures

1	A picture showing the differences in the same map from Diablo III	2
2	A picture showing the viewing angle in GTA 2	2
3	A picture showing the the stationary camera from Asteroids	3
4	A gantt table showing the estimated time shcedule of the project.	4