



UNIVERSITY OF AGDER

Left[2D]ie

by

**Jonathan Luu, Cecilie Gjerde Vågenes, Eivind Hystad,
Magnus Solhaug, Morten Suvatne Øvrebø**

Project report for DAT220, Fall 2016

Supervisor: Christian Auby

Faculty of Engineering and Science
University of Agder

Grimstad, December 2016

Status: Final

Keywords: SFML,

Summary: Her kommer oppsummeringen av prosjektet

Preface

This report is a result of the work accomplished by Jonathan Luu, Cecilie Gjerde Vågenes, Eivind Hystad, Mangus Solhaug and Morten Suvatne Øvrebø in subject DAT220 (Software Development 2) at University of Agder.

Grimstad

December 15, 2016

Jonathan Luu

Cecilie Gjerde Vågenes

Eivind Hystad

Magnus Solhaug

Morten Suvatne Øvrebø

Contents

List of Figures	i
List of Tables	ii
1 Introduction	1
1.1 Background	1
1.2 Problem Definition	2
1.3 Assumptions and Limitations	3
1.4 Literary Theory	3
1.5 Problem Solution	3
1.6 Report Structure	3
2 Theory	4
2.1 SFML	4
2.2 SFGUI	4
2.3 Design Patterns	5
2.3.1 State Machine	5
3 Process	6
3.1 Project Organization	7
3.1.1 Meetings	8

3.1.2	Tools	8
3.2	Priorities	8
3.3	Changes	8
4	Game Design	9
4.1	Description	9
4.2	Menu	9
4.2.1	Main Menu	9
4.2.2	Play	9
4.2.3	Highscore	9
4.2.4	Settings	9
4.3	Gameplay	9
5	Implementation	10
5.1	Platform and Libraries	11
5.2	State Machine	11
5.3	Rendering	11
5.4	State Play	11
5.5	State Menu	11
5.6	Objects	11
5.7	Collision Detection	11
5.8	Movement System	11
5.9	AI	11
5.10	Attack	11
5.11	Weapon	11
5.12	Player	11
5.13	Zombie	11

6	Testing	12
6.1	Procedure	12
6.2	Known Bugs	12
7	Discussion	13
7.1	Work Breakdown	13
7.2	Challenges	13
8	Conclusion	15
8.1	Future Work	15
8.2	How-to Latex Guide - Jonathan	15
	References	16
	Appendices	17
A	Dictionary and Abbreviations	18
A.1	First section	18
B	Title of Appendix B	19
B.1	First section	19

List of Figures

3.1	Oh really?	7
7.1	Pussy	14

List of Tables

1.1 Windows Versions	1
--------------------------------	---

Chapter 1

Introduction

1.1 Background

A platform game, or platformer, is a video game where the player lead a main character through one or several tracks. These tracks consists of several obstacles one have to pass and platforms one have to jump between.

The first platform games originated in the early 80's, but became popular first in the middle of the 90's. At one point, platform games were the most popular genre within video games. Between 1/4 and 1/3 of console games were platform games.

In this project, all the group members were born in the early-mid 90's, and have therefore grown up with platform games. The group have got a lot of inspiration from the game "Left for dead".

Windows	Win95	Win98	Win2000	WinXP	WinVista	Win7	Win8	Win10
----------------	-------	-------	---------	-------	----------	------	------	-------

Table 1.1: Windows Versions

1.2 Problem Definition

The main demand in this project was that it had to be programmed in C++. It was also a demand to use Jira for time-logging and setting up sprints. It was also a demand that Git was used for version control. Beyond this, the group stood pretty free to choose what they wanted to make.

It was decided to make a platform game. The game would have realistic physics, the enemy would have AI's, there would be weapons in the game. The group also decided to add hats as a cool, unusual feature in this game. This game was going to be very fun and addicting to play. To solve this problem, the player needed to have the opportunity to lose and win. To win the player has to kill all the zombies that appears, and to lose the player has to get killed by the zombies.

For the best possible execution of the task it was considered which main-classes that had to be included, and which under-classes these could have. After that it was decided which classes that had to be made first in order to make the other classes that would be addicted to them. The classes were added in Jira, and under the project-period there were added more tasks there as new ideas came up. Each member of the group chose their tasks from Jira, and assigned to new ones when they were done with the previous tasks.

After the project-period ended it was expected to have a working platform-game where the player have to kill zombies in order to win the game. The game has to be

1.3 Assumptions and Limitations

1.4 Literary Theory

1.5 Problem Solution

1.6 Report Structure

Chapter 2

Theory

2.1 SFML

SFML (Simple and Fast Multimedia Library) is a fast, a simple interface to the various components of your PC, to ease the development of games and multimedia applications [3]. It consists of five modules: system, window, graphics, audio and network. SFML is also multi-platform, which means applications made in SFML can compile and run on the most common operating systems: Windows, Linux, Mac OS X and soon Android and iOS.

2.2 SFGUI

SFGUI (Simple and Fast Graphical User Interface) is a fast, simple and native C++ GUI library for SFML [4]. This is a open source library, which means it allows you to use the library both in open and closed source projects for free. The library has a set of widgets, like window, button, label, scrollbar and more. These widgets can be styled using special properties built in SFGUI, which almost looks like the syntax of CSS.

2.3 Design Patterns

2.3.1 State Machine

Chapter 3

Process

Within software engineering, programming (the implementation) is regarded as one phase in a software development process.

There is an ongoing debate on the extent to which the writing of programs is an art form, a craft, or an engineering discipline. In general, good programming is considered to be the measured application of all three, with the goal of producing an efficient and evolvable software solution (the criteria for "efficient" and "evolvable" vary considerably). The discipline differs from many other technical professions in that programmers, in general, do not need to be licensed or pass any standardized (or governmentally regulated) certification tests in order to call themselves "programmers" or even "software engineers." Because the discipline covers many areas, which may or may not include critical applications, it is debatable whether licensing is required for the profession as a whole. In most cases, the discipline is self-governed by the entities which require the programming, and sometimes very strict environments are defined (e.g. United States Air Force use of AdaCore and security clearance). However, representing oneself as a "professional software engineer" without a license from an accredited institution is illegal in many parts of the world.

Another ongoing debate is the extent to which the programming language used in writing computer programs affects the form that the final program takes. This debate is analogous

to that surrounding the Sapir-Whorf hypothesis in linguistics and cognitive science, which postulates that a particular spoken language's nature influences the habitual thought of its speakers. Different language patterns yield different patterns of thought. This idea challenges the possibility of representing the world perfectly with language, because it acknowledges that the mechanisms of any language condition the thoughts of its speaker community.

3.1 Project Organization

In 1843, Ada Lovelace, a British mathematician, published an English translation of an Analytical Engine article written by Luigi Menabrea, an Italian engineer. To her translation, she added her own extensive notes.



Figure 3.1: Oh really?

3.1.1 Meetings

3.1.2 Tools

3.2 Priorities

Jada...

3.3 Changes

Chapter 4

Game Design

4.1 Description

4.2 Menu

4.2.1 Main Menu

4.2.2 Play

4.2.3 Highscore

4.2.4 Settings

4.3 Gameplay

Chapter 5

Implementation

5.1 Platform and Libraries

5.2 State Machine

5.3 Rendering

5.4 State Play

5.5 State Menu

5.6 Objects

5.7 Collision Detection

5.8 Movement System

5.9 AI

5.10 Attack

Chapter 6

Testing

6.1 Procedure

6.2 Known Bugs

Chapter 7

Discussion

7.1 Work Breakdown

7.2 Challenges



Figure 7.1: Pussy

Chapter 8

Conclusion

8.1 Future Work

8.2 How-to Latex Guide - Jonathan

See sourcecode

- This character _
- This character %
- Cite [\[1\]](#)
- Ref ??. Same with tables
- `#include <SFGUI/SFGUI.hpp>`
- `<hostname>`
- `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft`

References

- [1] S. Bebbington, (2016). What is coding?. Date retrieved 22.11.2016, from <http://yearofcodes.tumblr.com/what-is-programming>.
- [2] S. Bebbington, (2016). What is computer programming?. Date retrieved 22.11.2016, from <http://yearofcodes.tumblr.com/what-is-coding>.
- [3] L. Gomila, (Unknown). SFML. Date retrieved 06.12.2016, from <http://www.sfml-dev.org/>.
- [4] Unknown, (Unknown). SFGUI. Date retrieved 06.12.2016, from <http://sfgui.sfml-dev.de/>.

Appendices

Appendix A

Dictionary and Abbreviations

A.1 First section

Text of Appendix A is Here

Appendix B

Title of Appendix B

B.1 First section

Text of Appendix B is Here