

Computer Games Development

Project Report

Year IV

**Mantas Zalnierius**

**C00241626**

**28/10/2021**

# Declaration

# This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.

**Contents**

[Acknowledgements 2](#_Toc54714373)

[Project Abstract 2](#_Toc54714374)

[Project Introduction and/or Research Question 2](#_Toc54714375)

[Literature Review 2](#_Toc54714376)

[Evaluation and Discussion 3](#_Toc54714377)

[Conclusions 3](#_Toc54714378)

[References 4](#_Toc54714379)

[Appendices 4](#_Toc54714380)

# Acknowledgements

# Project Abstract

This project is to make a 2D platformer game with a custom physics system using robust collision detection in SFML.

The goal of the project is to implement a 2D physics and collision library that anybody can use to make their own 2D platformer games. For the physics aspect of my project, I will use ADSR envelopes to have a immersive symmetrical movement. For the 2D collision library I would use algorithms such as AABB (Axis-Aligned Bounding Box), SAT (Separating Axis Theorem) and Diagonals algorithms. I will also incorporate techniques that will make these expensive mathematical operations fast and responsive. I will also use different programming patterns such as finite state machine, factory and command. So that my project will have a very standardized codebase that anybody will be able to understand and use.

At the end of the project what I will expect to have done, is fully done game containing 2-3 Levels that use all the Collisions and physics that I have mentioned to have a game that feels immersive when playing the game.

# Project Introduction and/or Research Question

The reason why I choose this project idea was simply due to the fact that I want to have a game that just doesn’t look good but feel good to the player. As there are many games where the visuals are very impressive but the actual movement feels sloppy that discourages the player from enjoying the game.

There are multitude of different ways of doing collision in a 2D game. One of the most used algorithms is AABB. The reason for this is because it’s rather easy to implement and not many maths operations are used in AABB. The problem with AABB collision is that it cannot handle collisions between rectangles that are rotated which is a big draw back. That’s where SAT comes in. This algorithm is able to correctly handle collision detection between rotated rectangles. The issues with SAT are that it handles collision using the normal’s between each face of shape and through this it creates problems where if a shape has many faces which in result will cause a lot of computational time.

In respect to physics, I will be using an ADSR (Attack, Decay, Sustain, Release) envelope, where each letter will stand for stages in movement for a character in a video game. In the attack stage the character needs to overcome static friction to actually move. The Decay stage is where the character is slowed down due to kinetic friction. Then on the Sustain stage the character will be able to move for sometime until he gets to the relapse stage where the character will slow down to a halt. This leads to very symmetrical movement that feels immersive to the player.

The main idea of this project is to incorporate the 2D collision and physics I have mentioned that will lead to a game where the player will feel immersed.

Questions to answer

* How can collision be managed when there are hundreds of objects in the game at once?
* How will my game perform in environments where the machines are poor?
* Are there better Collision/Physics algorithms that do the same thing but at a faster computational time?

# Literature Review

# Evaluation and Discussion

**Project Milestones**

**Milestone #1**

Build a game in SFML with a game loop working (28/10/2021)

* Set up game using SFML\_SDK environment variable.
* Have assets folder set up containing fonts images etc.

**Milestone #1**

**Major Technical Achievements**

**Project Review**

# Conclusions

**Future Work**

# References

# Appendices