

College of Engineering and Informatics

B.Sc. (CS&IT) CT413 – Final Year Project 2019/2020

**Project Definition Document**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Project Name** | **Project ID** |
| Irish Senthilkumar | 16342613 | Soft Body Physics | SR6 |

Table Of Contents

List Of Figures

List Of Tables

Glossary

UE4 Unreal Engine 4

AI Artificial Intelligence

3DMS 3D Modelling Software

Introduction

In real life, objects are not completely rigid. When given enough force, they will deform to absorb the force being applied to them. Steel beams bend before breaking; jelly gets squished before separating; flags wave during a breeze. These dynamics are important to incorporate, since they increase the realism and immersion of computer graphics such as animated movies and video games.

Soft body physics is a recent field in computer graphics that focuses on realistically defining the motion of soft bodies. At a computer graphics level, these are scene objects whose points do not maintain a fixed distance from each other. At the fundamental level, all scene objects are comprised of polygons, which in turn are made of vertices which are connected by edges. These polygons can be combined in different combinations to produce a mesh, and this mesh can be manipulated to simulate soft body physics.

This project specifications document outlines the project and my proposals on tackling soft body collisions in Unity. This document is broken down into X sections…..

Deformation

2.1. **Introduction**

In real life, no object is completely rigid. Under strain, an object will deform to absorb the force applied on it. An object can deform in many ways, but there are two types of deformation: elastic and plastic. This section outlines the science behind deformation and discusses key concepts that will need to be accounted for in computer graphics in order to present a realistic soft body simulation.

2.2. **Elastic and Plastic Deformation**

For tensile stress (forces that pull on an object), elastic deformation is caused by low stress on an object, where a temporary change in shape is observed due to the molecular bonds being stretched but not broken. The shape of the object returns to normal after the stress is removed. However, if a large enough stress is applied to an object, the object will undergo plastic deformation. At a microscopic level, this deformation results in the molecules breaking their bonds and sliding past each other. Elastic deformation can be represented using Hooke’s law as this is similar to a spring system, but only when the elastic deformation is linear. Plastic deformation is non-reversible, so the object will never return to its original shape when the stress is removed. The elastic and plastic deformation stages can be represented on a stress/strain curve. As seen in Figure *2.*1, strain and stress are proportional to each other up to the yield strength. Elastic deformation is observed up to the yield strength point, and afterwards the object undergoes plastic deformation until the fracture point is reached. Upon reaching the fracture point, the object breaks apart. The ratio of elastic and plastic deformation is unique for all materials. Rubber bands have a large elastic region and a small plastic region, while glass has a small elastic region and a large plastic region.

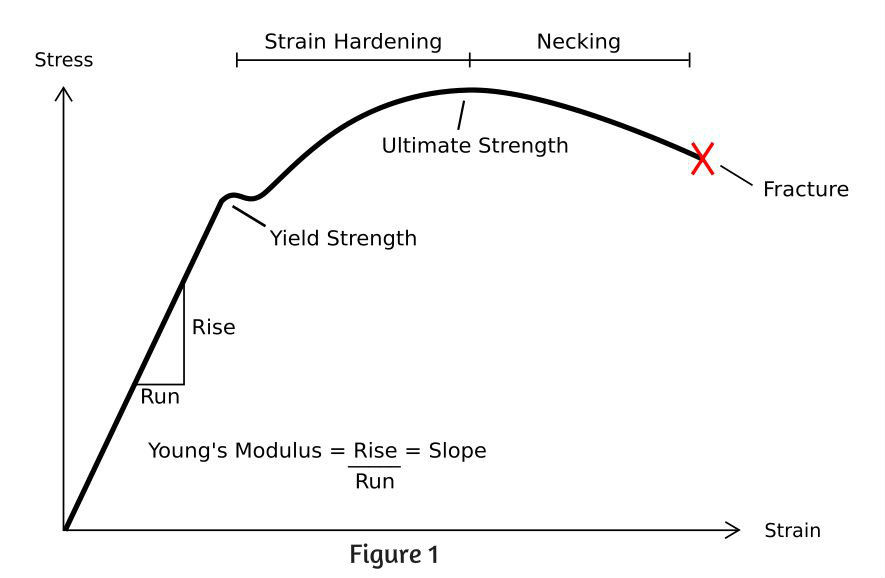


Figure 2.1: The stress/strain graph for an object.

Compressive stress (forces that push on an object) is the opposite of tensile stress, and objects display different behaviours under each type of stress. Some materials such as steel, can withstand high tensile stress and compressive stress. Other materials such as concrete, can withstand very high tensile stress but break apart during low compressive stress. When an object exerted to shear, bending, and torsion stresses, both tensile stress and compressive stress coexist.

Chapter 2. Technologies

2.1. **Introduction**

TODO

2.2. **Physics Engine**

Physics engines are computer software which provide an approximate simulation of physical systems. Soft body dynamics is not natively supported in some physics engines, so many physics engines were considered for use in this project. This section outlines the different physics engines which were considered, their features, their disadvantages, and their support for soft body simulation.

The first physics engine considered was Unreal Engine 4, developed by Epic Games. UE4 contains a complete suite of development tools designed for real time technology. UE4 is written on C++, a cross-platformed language that can be used to create complex, high performance applications. UE4 has many features, including blueprints, post-processing, advanced artificial intelligence, and a marketplace ecosystem. However, UE4 is much more suited to long term projects such as triple A games, rather than small projects such as this final year project. UE4 does not natively support soft body simulation, but plugins can be downloaded from the marketplace which can perform this task.

The second physics engine considered was Lumberyard, developed by Amazon. Lumberyard is based on Crytek’s CryEngine, a proven engine used for video games such as the Far Cry series. Lumberyard is written on C++ and Lua, a lightweight, high-level, multi-paradigm programming language designed for embedded use. Lumberyard has many features such as integration with Amazon Web Services for multiplayer and streaming features, a convenient real time lighting engine, and incredible graphics capabilities. The biggest drawback of Lumberyard is that this engine is still in beta testing, so the userbase is small and the support is minimal. Lumberyard does not natively support soft body simulation.

The third physics engine considered was Unity, developed by Unity Technologies. Unity is a popular cross-platform games engine which supports both 2D and 3D development. Unity is written in C#, a highly popular language that is simple, modern, and general purpose. Unity has many features, including customisable editors, post-processing, artificial intelligence pathfinding, and a marketplace ecosystem with a vibrant community. However, Unity has certain drawbacks such as being closed-source, inefficiency for large-scale use, and a subpar graphics pipeline. Unity does not natively support soft body simulation, but it is very easy to develop or can be downloaded from the marketplace. Unity was chosen as the physics engine to model the project due to its ease of development and support. ADD MORE TEXT

2.3. **3D Modelling Software**

3D modelling software is computer software which aids the user in creating a mathematical representation of a 3D object. In this project’s context, 3D modelling will be used to create game object meshes which have a clean and consistent topology, which is crucial to realistically model soft body dynamics. Complex 3D modelling is not natively supported in most physics engines, including Unity, and therefore a separate 3DMS is required. Only two 3DMS were considered since other mainstream 3DMS were too expensive for use in this project. This section outlines their features and their disadvantages.

The first 3DMS considered was Autodesk Maya, developed by Autodesk. Maya is the industry standard for computer graphics, and is used in a number of companies such as Pixar and DNEG. Maya contains an incredible amount of features, such as hair, cloth, and particle simulation, but it is a very complex software that takes long to master. Due to the small timeframe for this project, Maya was not chosen for use.

The second 3DMS considered was Blender, developed by Blender Foundation. Blender is hugely popular in industry and hobbyists alike, due to its ease of development and open-sourced nature. Blender contains many features such as sculpting, decimation of polygons, remeshing, all of which are ideal for use in this project. Blender was chosen as the 3DMS due to its ease of development, numerous features, and the ability to import Blender models directly into the Unity physics engine.

Chapter 3. Utilised Software