2-Dimensional Physics Simulation Environment (Sandbox)

Software Design Document

Branden Stahl

11/08/2023

CS 225, Fall 2023

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

**Introduction:**

The project is a 2-dimensional (2-D) physics engine. The goal is that it can be interacted with by the user. There will be a Graphical User Interface (GUI) that the user can interact with that will show a User Interface (UI) version of the physics simulation. There will be a gravity that can be toggled between using global (base earth gravity of 9.81m/s2 (1)) or particulate gravity (gravity based around the weight and distance of the particles). The reasoning behind this project is to help better understand how physics operations work, but also provide entertainment of providing an interactable 2-D physics simulation, known as a physics sandbox. This document will explain what the requirements of the program are (Table 1) and will list several test cases to test those requirements (Table 2).

**Background information:**

DAT files are a common type of ‘Comma Separated Values’ (CSV) file.

Known physics formulas in use for this project:

Gravitational Constant: (1)

Force between two gravitational bodies: (2)

Acceleration due to gravity: (2)

Force: (2)

Law of Cosines: (3)

Inelastic Collision Formula: (4)

**Requirements:**

Table 1: Requirement Specifications

|  |  |
| --- | --- |
| **ID** | **Requirement Specification** |
| 1 | As a user I want to be able to create new shapes. |
| R1. The user will be given a menu on right click to spawn in a shape of whatever type when not right clicking on a shape. |
| 4 | As a program designer I want to simulate 2d physics in an appropriate and realistic way. |
| R4. The outcomes of the program are mathematically **provable** to be correct following known physics formulas. |
| 5 | As a program designer I want to provide different shape options to the user. |
| R5. The user has the ability to create different shapes. |
| 6 | As a program designer I want to provide an accurate GUI layout that ensures the program can be navigated easily. |
| R6. There will be a comprehensive UI that will allow the user to interact accordingly with the program. |
| 9 | As a program designer I want to be able to easily edit various constants and settings for the program externally from the code. |
| R9. There will be 2 DAT files embedded within the project that will be able to store data for the programs use. |
| 10 | As a user I want there to be a menu for interacting with the entire program. |
| R10. There will be a ‘main menu’ screen that allows the user to edit settings and chose to launch either from a file or new. |

**Test Cases:**

I am at work trying to complete this document. I cannot implement test cases. I can just demo that my code works.

Table 2: Test Cases and Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req’t**  **ID** | **Test**  **Case**  **ID** | **Initial**  **Conditions**  **And Input** | **Expected Behavior**  **Or Output** | **Actual**  **Behavior**  **Or Output** | **Pass**  **Fail** |
| 1 | 1 |  |  |  |  |
| 1 | 2 |  |  |  |  |
| 1 | 3 |  |  |  |  |

Note: All test cases require debug mode to be ON!

**References:**

1: COMSOL Physical Constants Reference <https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/comsol_ref_definitions.12.025.html>

2: United States Naval Academy Physics Chapter 13 PDF  
<https://www.usna.edu/Users/physics/finkenst/homepage_files/SP211/Chapter_13.pdf>

3: Wikipedia: Law of Cosines <https://en.wikipedia.org/wiki/Law_of_cosines>

4: Wikipedia: Inelastic Collision <https://en.wikipedia.org/wiki/Inelastic_collision>