Project Proposal

Animal’s revenge

420-203-RE

MAZE
Ethan Tran
Anton Lisunov
Zachary Tremblay
Mackenzie Rouchdy

VANIER
CÉGEP/COLLEGE

Computer Science and Mathematics
Vanier College

Sleiman Rabah

September 7, 2023

1 Introduction

1.1 Description

Animal's revenge is an immersive interactive graphical physics simulator centered around projectile motion. Upon startup, users are greeted by a home screen with a variety of buttons and menus, determining where they want to go and what they want to do. The main idea is to get users to launch a projectile at a structure that was built with a drag and drop system allowing users to have invoke their creativity and design unique structures. Through this potent user-interface, users will be able to easily choose between animals and objects to launch. The user will also be prompted to choose different launchers to create a more unique experience for each user. The customization in this simulation is what will take it above and beyond similar applications. Moreover, the application will act as a learning medium in which users can experiment with projectile motion. This will be made possible with the various parameters and customization ability the users will have. For instance, users will be able to change the launch angle of the projectile. This change will influence the path of the projectile thus enabling users to recognize the results of such parameters. The application will be comprised of many parameter trackers. These statistics will be enhanced with live graphs that track the motion as it is in the air. These details are important as it allows the user to understand and interpret what is on their screen.

1.2 Goals

In accomplishing this project, our team is hoping to:

- **Educate.** Our project's main objective is to help our users understand physics related concepts, predominantly projectile motion, and collisions, by allowing them to visualize the applied concepts in a graphical environment.

- **Simulate.** By enabling the user to choose the different factors of the thrown projectile, such as angle of launch, along with the complete freedom for the structure building, as long as it follows the laws of physics, the users will be able to create an almost infinite number of scenarios. This will enable them to understand the impact of the different factors present in the experiment.

- **Engage.** By implementing game-like features along with a lot of interactive parameters, we make sure that the users are entertained throughout their learning experience. This will make them stay longer on our platform and learn even more about the subject.

1.3 Implementation

In order to implement such a project, the following steps would need to be realized:

1. Project planning and prototyping: Setup project, GitHub collaboration workspace, wireframes, UML diagrams, establish clearly what will need to be done.

2. Task identification and separation: Breaking down project into smaller tasks, identify dependencies, and assigning team members specific tasks with deadlines in accordance to their expertise and skills.
3. Home screen implementation: A graphical user interface must be implemented in order to guide the user to the application's functionalities, and allow the user to interact with various parts and navigate around the application.
4. Data representation: Define data structures to represent objects (projectiles, obstacles, buildings) and their properties (mass, radius, size, etc.). Integrate data structures using FXGL's entity-component system (ECS) and establish a coordinate system for positioning objects, handle distances and displacements.
5. Physics simulation: Implement physics equations, such as projected kinematics formulas, Newton's laws of motion, air drag, etc.
6. UI Design and user interaction: Design an intuitive UI that allows users to adjust physical parameters through the use of sliders, buttons, checkboxes, etc. Implement controls for zooming, panning, rotating, etc. Users should be able to navigate between different scenarios, the home screen, and the simulation.
7. Visual enhancements: Implement additional features that allow users to better understand the simulation, such as orbital trajectories, trails, velocity vectors, etc.
8. Switching between scenarios: Implement various built-in scenarios that users will be able to load in to simulator to see interesting results.
9. Testing, debugging, refining: Conduct unit tests, and performance test to ensure that everything works fine and that individual components of the application function correctly with each other.

2 Technologies

In order for such a project to be achieved, the following platforms, libraries and technologies will be used in its implementation:

- JavaFX: UI Framework for Java
- FXGL: Superset of JavaFX that is used in order to render 2D graphics, manage textures, handle animations, etc. Will simplify development process through an entity-component system and game state management.
- JBox2D: Java physics engine that provides 2D rigid body physics simulation capabilities. Will be used to handle collision detection, rigid body dynamics, and other physics-related tasks.
- ControlsFX: Provides additional UI controls that enhances the base JavaFX application. Includes dialogs, notifications, and other advanced controls for enhancing user experience.

- FXML and CSS: FXML will be used as the markup language to declare the UI. CSS will be used for styling.
- Java Sound API: Will provide the necessary functionality for playing sounds to enhance user experience.

3 Resources needed

To enhance the realism, functionality, and user-experience of the projectile motion simulator, the following additional resources will be used:

- Graphics and texture resources: High-resolution images and texture maps will be used to represent planets and other celestial bodies to enhance the visual appeal of the simulator. The main source is OpenGameArt.
- Audio resources: Sound effects accompanying collisions and other celestial phenomena, as well as ambient music will be used to enhance user experience and make interactions more immersive. The main source of audio media will be OpenGameArt.
- Configuration files: Configuration files will contain data that store the default conditions for planets (such as mass, velocity), physical parameters (such as gravity and other constants), and other presets that the user can load into the simulation.

4 Features

Animal's will offer a wide range of features to demonstrate a physical reaction of object on each other in 2 dimensions:

- Projectile motion simulation in 2D : Users will be able to simulate projectile motion by launching different projectiles at buildings. This will be done by projecting kinematic equations on Ox and Oy plane, taking into account factors like initial velocity, angle of projection, and gravity.
- Newton's 3 laws of motion:
  - Law of Inertia(Launched object maintain their velocity until affected by gravity or collisions.)
  - Law of Force and Acceleration($F = ma$; It will calculate and display the forces and accelerations)
  - Action-Reaction Pairs(It will not let the object to enter in other object and will provide with realistic collision)
- Drag-and-Drop building customization: Users can create custom structures by selecting, dragging, and placing different building blocks on the screen. Will include a variety of object, including walls, platforms, blocks, obstacles, etc.

- Projectile Selection: Multiple different animals and objects will be offered as projectiles. The will all have their unique masses, sizes, etc.
- Parameter customization: Application will provide all necessary tools for modifying and changing:
  - Initial launch direction (angle)
  - Initial speed
  - Projectile mass
  - Projectile size
  - Force of gravity
  - Air resistance (drag), and wind speed
  - Size of building blocks
  - Mass of building blocks
- Save and load: Allow users to save their simulation state and reload it. Will be able to store user-designed structures for future use as well as a variety of built-in scenarios with various preset parameters and buildings.
- Additional UX features: modifying background image (look and feel), interface color, design, style, small micro-interactions (button hovers, clicks), loading skeletons, etc.
- Sound: Will include a soundscape for different scenarios in addition
- User-Friendly Interface: The GUI will be intuitive, with easy-to-use controls for adjusting parameters and navigating through the simulation.

5 Similar projects

The original inspiration for our project was the game Angry Birds. In that game, you use different kinds of birds, each with their own superpowers to attack monuments containing green pigs. The objective is the eliminate all the pigs by launching the birds and destroying their structure. The game is simply meant to be fun and relaxing as it progresses through different levels, each having prebuilt structures. Our project will be centered around a similar idea but will be more focused on being educational rather than fun. We will follow the laws of physics, something that angry birds does not always do, to make sure that our users learn about projectile motion among other physics related concepts. In addition, the users will be able to construct their own structures to be able to test an infinite number of scenarios as well as fully customize physics parameters such as gravity and air drag. Finally, we will display a lot of information about each and every throw made by the users to help them understand the physics behind the game.