Oregon State University
College of Engineering

CS Capstone Design Document

April 15, 2019

Interactive 2D simulations to support Inquiry-Based Learning in Mechanical Engineering

Prepared for

Oregon State University, School of Chemical, Biological, and Environmental Engineering

Milo Koretsky _____ Signature _____ Date

Tom Ekstedt _____ Signature _____ Date

Prepared by

Group53

Education Simulations

Cameron Friel _____ Signature _____ Date

Kelli Ann Ulep _____ Signature _____ Date

Samuel Wilson _____ Signature _____ Date

Abstract

This project involves solving the problem of some universities lacking resources to visually show physical and mechanical interactions for Mechanical Engineering concepts. This project is built on the research that shows that students can achieve a better understanding of difficult concepts by learning through interactive simulated environments. By implementing 2D simulations based on these concepts, students will be able to visually interpret the concepts in the course. The goal of this document is to document the design for these 2D simulations, explaining the implementations of the features which shall be expected for this project.

CONTENTS

1 OVERVIEW

This document contains an in depth description of the designs to be used in the 2D Simulations for Inquiry-Based Learning project. This document will go over the UI design layouts needed for the simulation web page, the implementation of webpage hosting, and the implementation of the animation. The components within each of these sections are also explained.

1.1 Scope

Our product will consist of six variations of a two dimensional simulation based on mechanical engineering concepts in order to support inquiry based learning for mechanical engineering students. The simulations will be hosted on the Concept Warehouse website, which is a website that provides interactive learning opportunities for classrooms. While the simulation is running, real time mathematical feedback will be provided in the form of graphs or simply values in the simulation. There will be start, stop, and reset button within each simulation so that the student can explore the simulations as many times or at as many points as they need to. The students will also be able to modify specific parameters within the sixth variation of the simulation, such as mass, friction, and angles.

1.2 Purpose

The purpose of this document is to present the design specifications for the Interactive 2D Simulations to support Inquiry-Based Learning in Mechanical Engineering project. This document will expand on the technologies used to implement the project, which will include:

- The process of serving the simulations to the student
- The Game Engine used to procure the simulations
- User Interface designs for the web page
- Data Visualization
- Software for creating the simulation assets

1.3 Intended Audience

The intended audience for this document is primarily for developers that will be working on the project and need to be on boarded. In addition to this, the document is used as verification to the client's of the project, Milo Koretsky and Tom Ekstedt, to make sure that the design falls inline with the expectations they have provided.

2 GLOSSARY

- Concept Warehouse - A platform for engineering students to answer questions inside or outside of class.
- User Interface (UI) - The aspect of a program that the user interacts with directly through keyboard and mouse.
- Simulation - A digital and visual representation of a physics model.
- Object - A particular instance of a class, where the object can be a combination of variables, functions, and data structures.
- JavaScript - A core scripting language used in the web. It is used to create interactive elements within a web page.
- Chart.js - A lightweight JavaScript library for creating interactive graphs and that is free to use for the web.

- Open-source - Software code that has been made available for public use and manipulation.
- Inkscape - A free and open-source 2D vector graphics editor.
- Defold - A free, 2D game engine, used by professional game developers.
- Coefficient of Restitution - ratio of the final to initial relative velocity between two objects after collision. Values range from 0 to 1
- Bootstrap - A toolkit for designing web applications
- Mockup - A prototype providing the design and functionality of a system
- Input Bindings - Project wide table specifying how input translates into actions in the script

2.1 Design Timeline

- Jan 7 - Jan 20: UI creation
- Jan 20 - Feb 20: Simulation creation
- Feb 20 - Mar 22: Graph Add-on and Bug Fixes
- Mar 22 - April 10: Clean Up and Stretch Goals
- April 10 - Expo: Expo Preparation

3 Design Viewpoints

3.1 Introduction

The different components of the web page is the hosting information, game engine, user interface, graphs, and art software. The following section explains each of the components and the implementation design.

3.2 Hosting

The project will consist of several files that will be delivered to the client. The client will then upload these files to their own server to be distributed to students. The end product will simply be a simulation that can be displayed to students. The simulation consists of 5 pendulum variation cases and an exploratory mode. The flow of the system is that the instructor dictates when students are able to view the simulation through Concept Warehouse. Then the server updates the content of the web page with a new case. The cases are shown in sequential order from 1 to 5. Exploratory mode, where users can modify parameters, is also unlocked by the instructor.

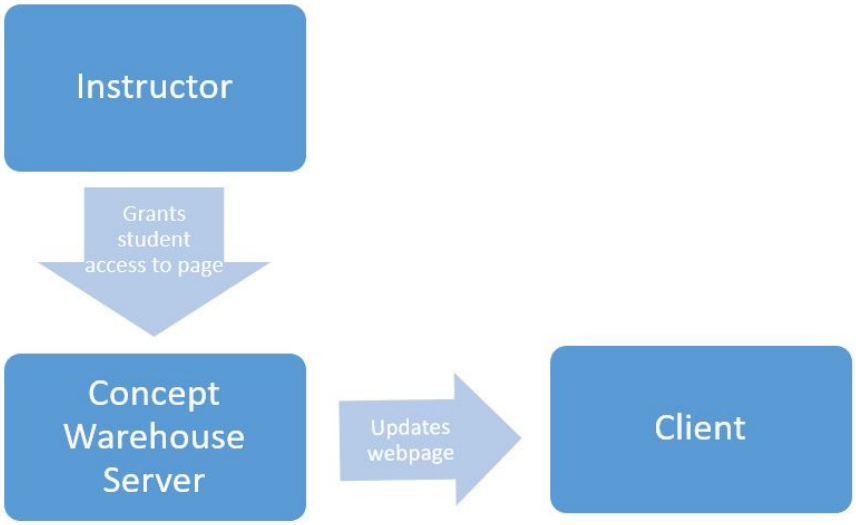

Fig. 1. Hosting Diagram

3.3 Viewpoint: Game Engine

3.3.1 Design Concerns

While game engines are useful for speeding up aspects of game creation, there is a risk that certain tools will not be usable for programs other than games. In this instance, the physics engine used may not work correctly with the projects simulations.

3.3.2 Design Elements

Game engines are programs which use an interface to allow easy creation and manipulation of UI and physics items. Certain functions that are common to game development, such as signals, are built-in. The main processes, which are unique to the game being created, are developed using scripts. These scripts are attached to individual elements of the game, though some can hold global variables. In the case of this project, the simulations being created are very similar to games. The team will use this similarity to their advantage by using game engines to quickly create physics simulations and UI.

3.3.3 Function Attribute

The game engine, Defold, will allow the team to quickly create multiple simulations of similar types. The ability to visually create the UI in real time will significantly increase production speeds, while maintaining the freedom of scripting.

3.3.4 Pendulum Animation

For the pendulum animation, the motion will be based on the following equation:

$$\theta'' = (-1 * g/L) * sin(\theta) \tag{1}$$

where g = gravity, L = pendulum arm length, θ = angle in deg
The angular acceleration of the pendulum is a constant of gravity divided by the length of the pendulum arm, multiplied by the sine of an angle. The velocity and position is changed over time. To account for air resistance or friction in the system, the angular velocity can be reduced by a constant each animation frame.

$$\theta' * k \tag{2}$$

Where k is a constant between 0 and 1 where 0 is a stationary pendulum, and 1 is no air resistance.

Defold includes a Box2D physics engine to simulate the physical interactions between two pendulums - for the cases that there are more than 1 weight in the pendulum system. The animation shall use the *collision object* feature.

3.4 Viewpoint: User Interface

The User Interface is the gateway to connecting with the end user. It allows for interactivity with the simulation.

3.4.1 Design Concerns

A disorganized UI or a UI that is hard to understand can impede the student's learning. Even if the animation runs well in itself, if the web page it is placed in is hard to manipulate, it can take away from the animation.

3.4.2 Function Attribute: Normal Mode

The user interface mockup of Case 1 is shown in Figure 2 and Figure 3. Figure 2 will be the initial screen shown. This will be the normal mode of case 1. Normal mode is applicable to all 5 of the variations of the impact pendulum scenario. The normal mode components with their description are as such:

- Start/Pause Button - Starts the simulation or pauses it
- Reset Button - Resets the simulation
- Graph of angle vs. time
- Live measurement updates - Updates the angle and height of each weight, velocity, and the time every 0.1 seconds of the simulation
- Starting measurements - Length (m), weight (g), angle (degrees), height of weight off the ground (m), and coefficient of restitution

The graph component design is explained in section 3.5.

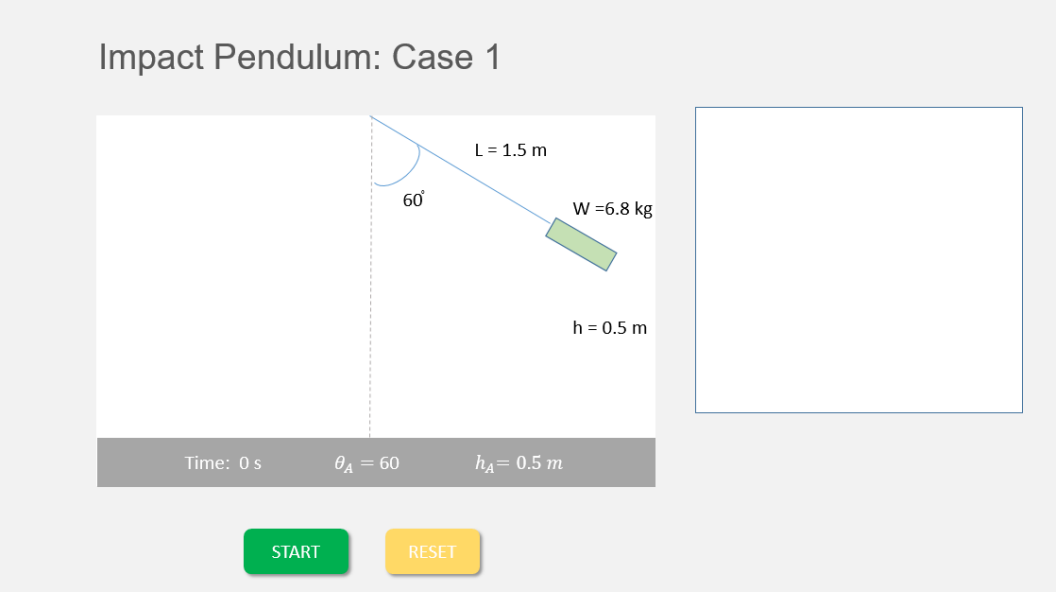

Fig. 2. Pendulum Case 1 - Before animation UI

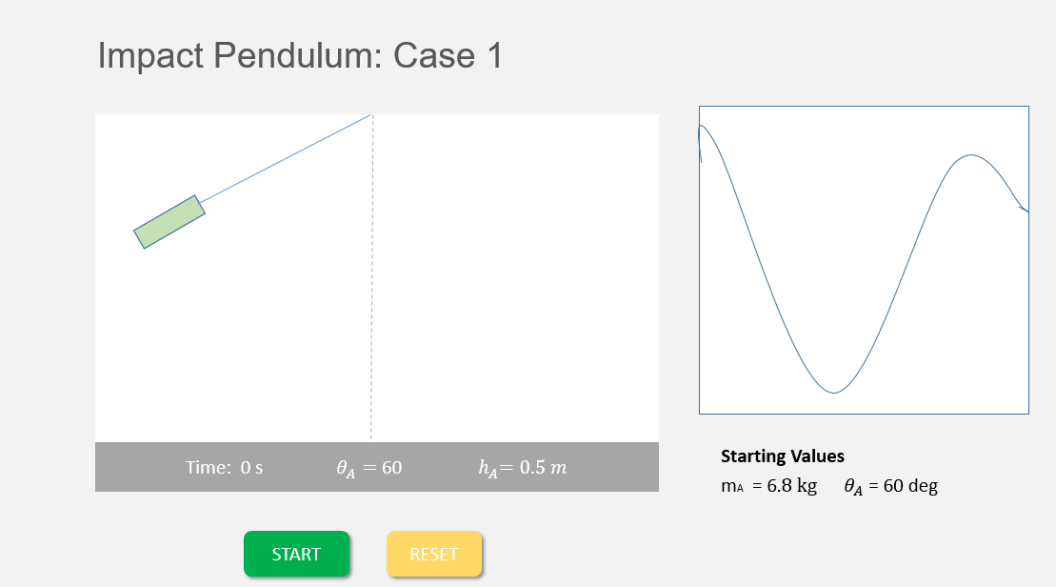

Fig. 3. Pendulum Case 1 - After animation completion UI

3.4.3 Function Attribute: Exploratory Mode

The mockup of the Exploratory mode where the user can modify parameters is shown in Figure 4 and Figure 5. The exploratory mode components are

- Start/Pause Button - Starts the simulation or pauses it
- Reset Button - Resets the simulation
- Graph of angle vs. time

- Live measurement updates - Updates the angle and height of each weight, velocity, and the time every 0.1 seconds of the simulation
- Option to choose between 1 weight or 2 weights
- Starting parameter input sliders for the angle (degrees), length (m), weight (g), and coefficient of restitution.

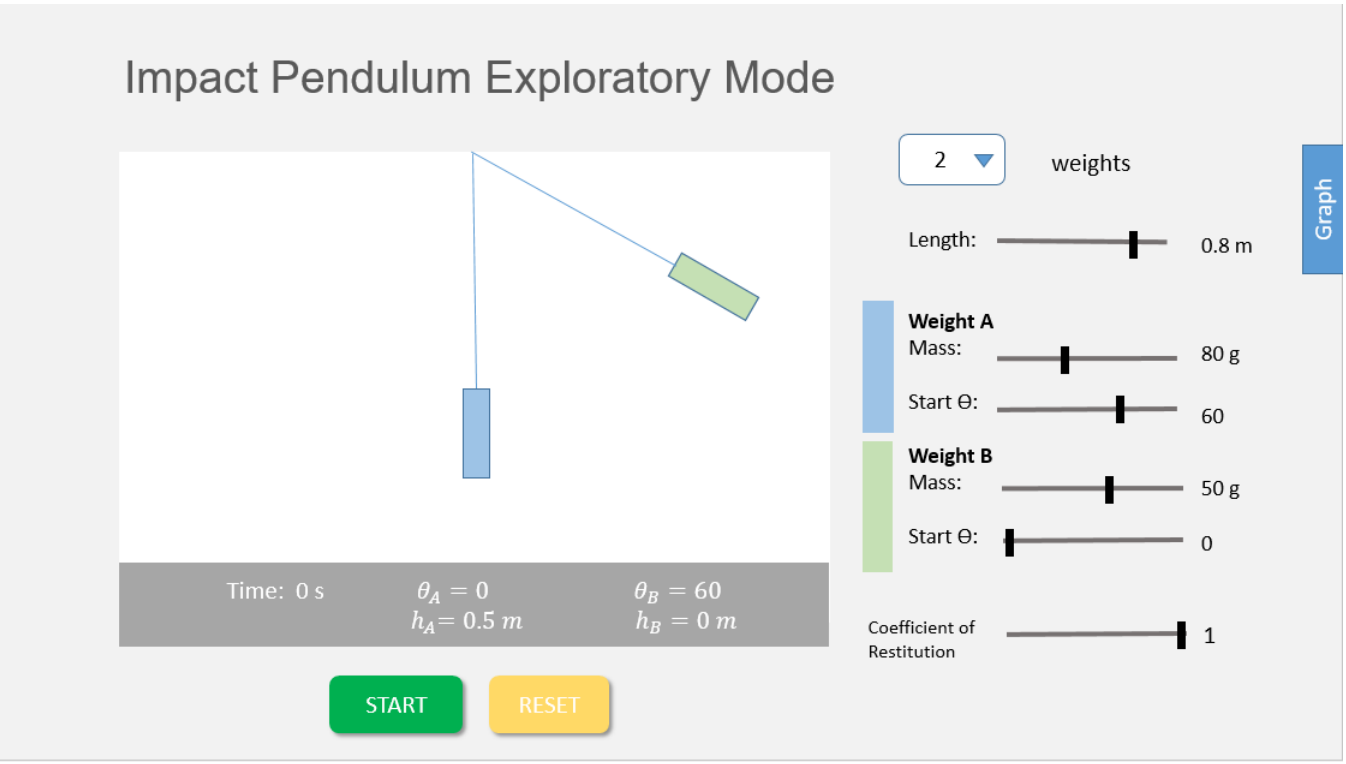

Fig. 4. Exploratory Mode Start Screen

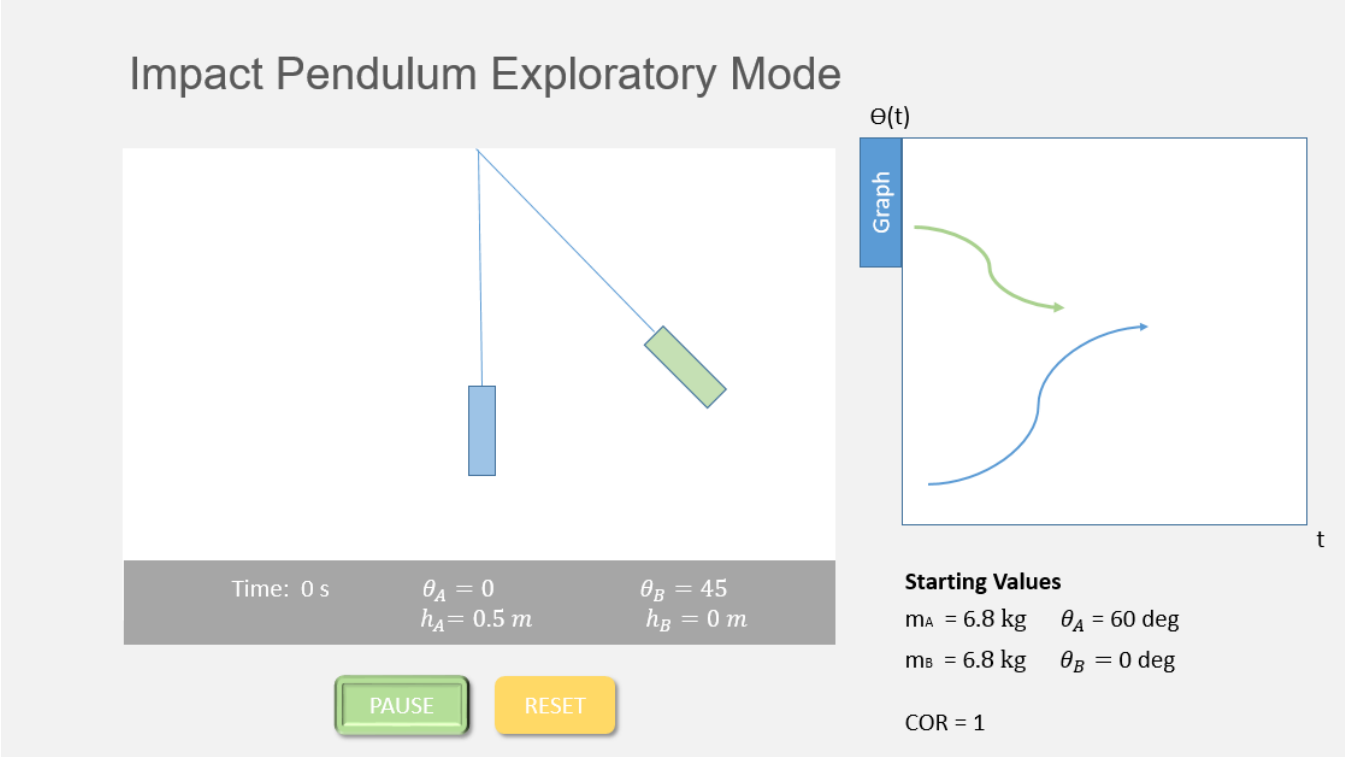

Fig. 5. Exploratory Mode Mid Animation

The game engine will construct the start/pause button, live measurement updates, and the input sliders and menu options. These components relate strongly to the animation.

Defold has an input system to capture values from the mouse or keyboard and then dispatch the raw inputs as *actions* to the script. The sliders, drop down menu, and the pendulum in the simulation can be told to listen for user input by sending the message "acquire_input_focus." The input bindings feature provides a table on how the input translates to the *actions*. Then this the action is implemented on the object with the on_input() function. The animation receives inputs from the input sliders and the drop down menu, so user input can be limited to reasonable measurements for the pendulum system. The values of length, mass, and starting angle will be passed into the pendulum motion formulas to affect the animation. The coefficient of restitution input will affect the pendulum game object's setting *restitution* - affecting how the objects react after colliding (i.e. sticking together or not). Defold has a *collision object* feature that has built in physical behavior and properties.

The graph of the weight's angle over time will replace the input sliders after starting the simulation. For styling of the UI, the Bootstrap library will be used in addition to the Defold canvas. Defold can build an application to run in a web page and can be provided with custom HTML and CSS.

3.5 Viewpoint: Graphing

3.5.1 Design Concerns

Graphs can be over stimulating if too much information is being presented to the viewer. Caution must be had when creating the graphs, so the layout is clear, and only important data points are being used.

3.5.2 Design Elements

Graphs will be displayed to the screen with Chart.js, which is a lightweight JavaScript library that is free to use. Once the library is included with the script tag, a graph can be included by creating a Chart object with modifiable fields to build up your graph, such as, label for the graph's title, data for the graph's data to be graphed, and backgroundColor. After the object is made, it can be referenced to a canvas tag in order to be embedded within the web page.

3.5.3 Function Attribute

Graphs will be used within the UI for data visualization to provide the student with another resource to understand the interactions within the simulation.

3.6 Viewpoint: Art Software

3.6.1 Design Concerns

The art involved in such dynamic programs is very important to conveying necessary information to the user. If the UI themes or graphics of a simulation are unclear, the user can miss interpret what is happening in the program, or lose interest in it due to poor quality. The color choices for the art themes of the interface are also extremely important, considering part of the user base will be blind to certain colors.

3.6.2 Design Elements

Using Inkscape, graphical representations for the elements of the simulations will be created to provide visible feedback to the user. These elements will be created in a way that makes it very clear to the user what they represent. The images provided in the User Interface section of this document give a good example of what these elements will look like.

3.6.3 Function Attribute

The art software chosen, Inkscape, will allow for the creation of simple and clear simulation graphics. Being open-source, Inkscape will also be available to everyone in the project team.

3.7 Conclusion

The core aspect of this project's development revolves around the use of the Defold game engine, which will be supported with the current UI designs, graphing scripts, and 2D images created with Inkscape.