CS Capstone Technology Review

June 11, 2019

Interactive 2D Simulations To Support Inquiry-based Learning in Mechanical Engineering

Prepared for

Oregon State University

Tom Estkedt, Milo Koretsky

Prepared by

Group 53
Learning Simulations
Samuel Wilson

Abstract

This document seeks to review several technologies, which will be integral in the creation of a project dedicated to creating a 2D simulation website. The 2D simulation will be used for the education of mechanical engineering students in simple physics. Examples would be: pendulums, pulleys, rolling cylinders, etc. Game engines, art software, and project management software are the technologies being reviewed. The goal will be to choose the most well fitting software for the project, based on a set of criteria.

1 INTRODUCTION

This project aims to solve the problem of some universities lacking resources to visualize physical and mechanical interactions for Mechanical Engineering concepts in the classroom. By implementing 2D simulations based on these concepts, students will be able to visually interpret the concepts in the course. The solution to be implemented is the creation of 2D simulations using client side Javascript. The student will be able to modify certain values like the mass of an object or the angle that an object is dropped within a simulation in order to understand how interactions that occur in the real world are rationalized. This document looks over technology that can be used in the project's creation, in order to decide the best options. Game development engines will be compared, along with 2D art software and project management software.

2 GAME DEVELOPMENT ENGINES

The main feature of this project requires 2D physics simulations, along with several important UI controls. A possible solution to building this would be to use game development engines. When choosing one, the key aspects to look for are:

- Number of features
- Ease of exporting
- Clear user interface
- Easy coding language to learn
- Quality of Physics implementation

2.1 GDevelop

The first engine is GDevelop, which is an open-source game creator with an online UI. This engine seems to have all necessary features, though simple. The website promises web exporting with one click. Its designed for 2D games, so sprites and particle effects are implemented well. The UI can be accessed in a browser, as well as in a desktop app, which both are very simple, with a focus on visual programming. Objects in this engine can also use a built-in physics system with the ability to change friction, mass, and forces like gravity. Overall, this engine has a solid set of features, but they are simplistic and the visual programming does not seem intuitive. [1]

2.2 Defold

At first glance, Defold seems to be more feature complete then GDevelop, possibly due to its use by professional companies. Defold allows for easy development to the web using HTML5, also having a single click export. The software is only available on desktop, with no browser development. Defold also promises easy tools for version control. The UI for Defold is much more robust than GDevelop, with the layout of a typical IDE. Lua is used for scripting, which is a lightweight language with some influence from C++, so there should be a small learning curve for our team. Finally, Defold uses Box2D [2] for its physics, which is a well known C++ physics engine. Defold seems like a very promising option, with higher quality features than GDevelop. [3]

2.3 Godot

Godot is an open-source, desktop application. It has a similar look to Defold, with a professional layout and set of features. Godot doesnt seem as clear on whether its web exporting is quick and easy. As stated, the UI for Godot is clear and has a well organized layout. This engine uses a custom language called GDScript, which is based on Python. It also has the ability to use visual scripting. Generally, Python is a very simple language, but some learning may be required. While not promoted as a feature, it does seem to have a physics engine based on Box2D. Godot seems to be similar in features to Defold, though the custom programming language may get in the way of quick production. [4]

3 2D ART SOFTWARE

In order to give feedback for the physics simulations, there needs to be a visible representation of the model. To do this, the team will need graphics software to create simple 2D art. In order to decide which software to use, the following criteria will be used.

- Software purpose
- Image manipulation
- Clean interface
- Ease of shape creation

3.1 Inkscape

Inkscape is a vector-based graphics software, which means objects created in the program can be easily scaled and modified. It seems to be highly recommended for making logos. It lacks in tools for manipulating images that were imported, compared to other software. Inkscape has a good interface for quickly making images which are designed with shapes. This project will only need basic shapes and lines for each element of the simulation, so Inkscape seems like a very good option. [5]

3.2 Krita

Krita is designed for digital artist and illustrators. It has very few features that revolve around image manipulation. It’s expected that the user will be an artist who will want to customize everything. Most of the interface is moveable to function as an efficient canvas. These features are not very important for the project, unless the quality of the art needs to be improved later. Krita has some functions for creating images from shapes, but its functions are clearly for more artistic pursuits. Overall, Krita would be a good option if more detail was needed for the project, but based on the current criteria its too complex. [6]

3.3 GIMP

GIMP advertises itself as an open source, image manipulation program. Its tools focus on modifying imported photos and art with a large variety of graphical effects. This makes it great for creating textures and interesting graphics, however GIMPs tools may be more than what is needed. Due to the vast array of the tools and effects, GIMPs interface is not as well put together, especially compared the previous two options. For creating simple graphics made of shapes, GIMP seems overly complex, though it has the features to support it. For the purpose of this project, GIMP seems to more than what is needed. It is also for more general purpose image editing; not having a specific design goal, when compared to the other options. [7]

4 Project Management Software

In order to keep the project on time, and organize the tasked assigned to each member of the team, our group will be using project management software. The criteria for choosing the best choice are:

- Designed for small groups
- Feature Variety
- Ease of task management
- Integration with other software

4.1 Clubhouse

Clubhouse follows a Kanban board style, using a ’Story’ for each task to complete. Clubhouse has a simple interface and flow, although it lacks some features that the other options have. It is designed for Agile development, which may be important, depending on the team’s design goals. This software only allows three people in a project when using the free version, but that shouldn’t be a problem with our team size. Finally, Clubhouse has the ability to connect with Github and Slack, which our team uses regularly. [8]

4.2 Pivotal Tracker

Similar to Clubhouse, Pivotal Tracker focuses on Agile development and uses ’Stories’ to designate tasks. It is not specifically designed for small groups, but the free version is only limited to three people per project. A key feature of Pivotal Tracker is the analytic tracking to show progress for members and the team as a whole. This would be useful to keep each member accountable. Task management is simple, using Kanban boards, and the interface is clean and organized. Pivotal Tracker’s integration is extensive, including the teams most used platforms, Github and Slack. [9]

4.3 Redbooth

Redbooth promises good organization for both small and large groups. It uses a Kanban board, like the other options, but it does have some features that the others don’t. One of the features that the team would be most likely to use is the Gantt chart integration. The other options don’t seem to have this basic function. Redbooth is just as easy to use as the other options, but with more functionality. Oddly, Redbooth does not have Github integration, though it does have Slack and Box integration. Our client wishes to use Box, so it could be a fair trade off for the Github integration. [10]

5 Conclusion

After analyzing the technologies of game development engines and 2D graphics software, there are pretty clear choices for the best in each category. For game engines, the goal is to find something robust, with accurate simulations, so the amount of problems will be minimal. The team should not run into something that is not supported by the engine. It should also be easy to script in the engine, with little learning needed. Both Defold and Godot have been used by professional developers, and were released to the public. They both have a good set of features, but Godots custom language may have more a learn curve for the team, so Defold will be the first choice for this technology category.

For the choice of 2D graphics software, the goal is the opposite of game engines. The project needs a very simple and quick image creation tool. Krita and GIMP have a wide set of tools, but between them, Krita seems to have a better user interface. Unlike these two, Inkscape fits the projects need for simplicity. It also has a clear user interface with just enough tools for the teams needs. Unless more interesting art is needed, Inkscape is the best choice for this project.

Finally, for the choice of best project management software, the options are all very similar. All of them use Kanban boards for organization, with some focus on Agile development. Redbooth has the most features out of the choices, with the addition of some very useful ones. All of the software choices had very clean interfaces, so there is not clear winner for this criterion. For integration, Redbooth was missing Github, but the inclusion of Box integration somewhat negates this flaw. Overall, Redbooth promises the most usefulness out of the choices, despite all of them being very similar.

References

[1] Gdevelop. [Online]. Available: https://gdevelop-app.com/

[2] Box2d. [Online]. Available: https://box2d.org/

[3] Defold. [Online]. Available: https://www.defold.com/

[4] Godot. [Online]. Available: https://godotengine.org/

[5] Inkscape. [Online]. Available: https://inkscape.org/

[6] Krita. [Online]. Available: https://krita.org/en/

[7] Gimp. [Online]. Available: https://www.gimp.org/

[8] Clubhouse. [Online]. Available: https://clubhouse.io/product

[9] Pivotal tracker. [Online]. Available: https://www.pivotaltracker.com/features

[10] Redbooth. [Online]. Available: https://redbooth.com/features