



CLARA'S 3D WORLD

Project Proposal

Abstract

This document explores 3 alternative Java-script Frameworks to potentially develop Clara's 3D world. This document also lists the High level business functions, use case list of each business function and the proposed development release schedule for the project.

Intended for Jonothan Weightman

Prepared by Group PA2207:

- Jarrod Baker (20487821)
- Djed Curtis (19744647)
- Hasan Karabork (19679091)
- Toufic Tannous (18528419)

Table of Contents

Executive Summary	3
Introduction	4
Problem Statement.....	5
Client Background	7
Project Background	8
Recommended solution: Babylon JS with External 3D models.....	9
Solution Description	9
Babylon JS.....	10
External 3D models.....	10
Storage.....	11
Business Case	11
Risks.....	11
Recommended solution summary.....	12
Alternative Solutions	13
Alternative Solution 1: Pixi JS	13
Solution Description	13
Business Case	13
Risks.....	14
Alternative Solution 2: three JS	15
Solution Description	15
Business Case	15
Risks.....	16
Alternative Solution 3: Developing Custom 3D Models.....	17
Solution Description	17
Business Case	17
Risks.....	17
Clara's World - High Level Business Functions	18
Use Case List.....	18
Development Release Schedule	19
Reference List	20

Executive Summary

The project complete aim is to provide the best alternative solution from the three which identifies which program will be used to make Clara 3D, which will fulfill the requirements of the system.

The three solution that the team have chosen is Babyloan.JS,3JS and PixiJS. As a team we will identify which solution will be the best approach that will be used to make Clara 3D. For the project proposal will involve choosing the best software dedicated to the needs of the client and the team.

The main factors that are going to be involved in the project will be based on the time to create the project, cost and the scope and quality of the final system.

For the project proposal it will aim to identify which software and what crucial tools that are needed to develop the project. It will help to enhance the high-level requirements set by the client which will help the team to choose the right project, so that it will improve the visual and animation of Clara's 3D World.

Introduction

The purpose of this document is to introduce the Clara's world 3D project proposal. Here we will provide a brief outline as to the purpose of the project and the problems being faced, then we will present our three alternative solutions, one of which we will recommend and our general approach to the solution along with when and how we will release these solutions.

Our main objective is to provide a new view to the already well-known Clara leaning tool used by various computer science and ICT related students at the University of Western Sydney.

As per our project brief, this new view will transform the current 2D representation of Clara to a 3D model utilising various technologies to incorporate this view into the browser.

Problem Statement

Clara's world is a game designed around learning how to write code. It was designed and developed at the University of Western Sydney so that its ICT and computer science related students would have an easy platform to develop and practice their coding skills without having to dive into something too unfamiliar.

Problems we have identified:

- The current state in which Clara's world has been depicted is only two dimensional, the main issue this 2D depiction promotes is that it is quite far behind the modern expectations of how games should look and feel, this may cause some students to not feel as immersed in the experience as they should be.

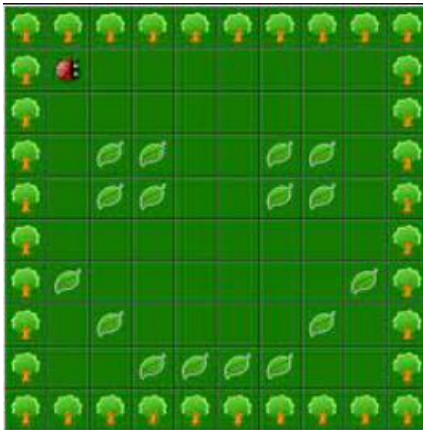
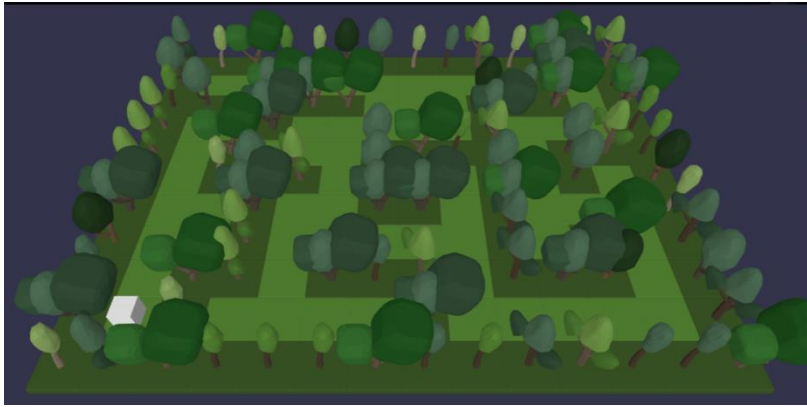


Figure depicting current state of Clara's world

Our goal is to transform the current two-dimensional depiction of Clara's world into something more inline with modern norms, namely making the game three dimensional. We believe that this portrayal of Clara's world is the next logical step in the evolution of this game.



*Figure depicting our vision
of Clara's world*

The main benefit from achieving our goal is to have students feel more immersed during their experience within Clara's world which may allow for their generation of coding skills to be greatly enhance

Client Background

Dr Tomas Trescak, our client for this project, possesses a Doctorate in Computer Science, with a specialisation in Artificial Intelligence. Dr Trescak coordinates the Information Technology and Communications course at Western Sydney University and is currently the lecturer of the Intelligent Systems unit.

Some of his interests in relation to IT include; 3D virtual worlds, Artificial Intelligence, Software Engineering and Human-Computer Interactions. The educational Clara's World programming game was originally created by Dr Trescak, and thus that is his connection to this project.

Project Background

The Clara's World game was designed to teach students the fundamentals of programming through a coding window paired with a user-friendly visual interface to display the outcome of the user's written code.

The game has many levels, varying in difficulty, that are released weekly to the users. Clara's World has been quite the success, as the first year Programming Fundamentals unit at Western Sydney University is based entirely around it.

Students are introduced to a simplified Java language environment, using Green Foot, that allows them to learn the basics of coding while getting useful visual feedback to assist their learning.

The creators of the game, Dr Tomas Trescak and Dr Anton Bogdanovych, have even released a Textbook titled "*Learning Java Programming in Clara's World*" to assist those who wish to use Clara's World.

Clara's World was entirely built using only ReactJS and TypeScript, however Dr Trescak built a custom compiler to translate the written Java to JavaScript which is based on JavaAST.

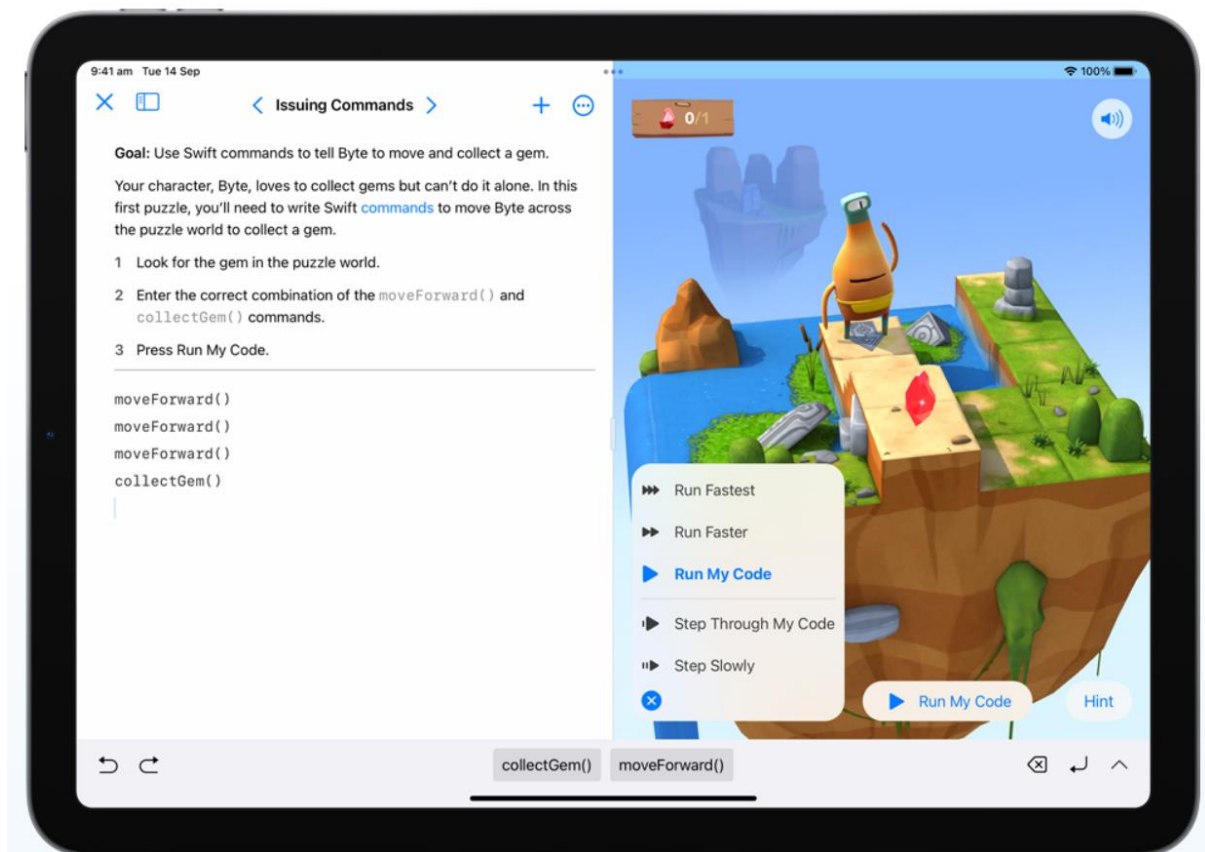
This project aims to develop a three-dimensional visual environment for the Clara's World game, replacing the current legacy two-dimensional environment.

Recommended solution: Babylon JS with External 3D models

Solution Description

Our client, Tomas Trescak has requested we use Babylon JS (a javascript framework) to implement Clara's 3D world along with 3D models imported from external vendors, and used by us in Unity (a 3d modelling/game engine).

The inspiration for the 3D modelling of Clara's world comes from an apple product called Swift Playgrounds. Swift playgrounds has largely the same functionality as Clara's world, it is an educational tool to teach programming to beginners, and the visuals shown below are the target for our project.



(Above photo sourced from: <https://www.apple.com/au/swift/playgrounds/>)

We aim to deliver the project without any cost, with the exception of buying 3D models from external vendors.

We are using Unity to modify and prepare our 3D models. The current process of preparing 3D models for use in Clara's 3D world includes:

- Opening the prefab files in a unity project
- Exporting those prefab files to fbx files using unity engine
- Convert those fbx files to glb files

(Note: prefab, fbx and glb mentioned above are all file types)

Once the files are in glb format, they are usable by Babylon JS for rendering in the browser.

With all the tiles converted to glb format, Clara's 3D world is generated dynamically using level information. This level information is currently being received via Json files, but may later be received from a database.

The whole system (code, level information, 3D models, other related files) will be stored on servers provided by the University of Western Sydney and deployed via these servers.

Babylon JS

Benefits:

- Free
- Open source
- Powerful/Versatile web based rendering engine
- Active community (helps for asking questions, also means Babylon js is still improving)
- Extensive documentation

External 3D models

Benefits:

- Ready for use out of the box
- Saves a lot of development time that might be spent on custom 3D modelling

Potential Downsides:

- May be hard to find different 3D models that follow the same theme/color scheme/design pattern, meaning 3D models from different vendors may not work together very well.
- Not enough variety to choose from in 3D models in the marketplace, providing a repetitive and bland user experience

Storage

Benefits:

- Storing the files and code on university servers, rather than in the cloud, will allow the client full privilege as to how the project is deployed.

Business Case

BabylonJS, as stated prior, is an extremely powerful web rendering engine, and due to it's active and helpful community, it's our choice for this solution. The BabylonJS API doesn't change as frequently as other engines, the developers are also very active, and the BabylonJS website provides a powerful testing tool called playground.

To add to these positives, large brands like Adobe (Bond, M. 2018) and Mercedes (Mercedes. 2021) have used BabylonJS before, and employees from Sony have also given BabylonJS positive feedback (Bawel, D. 2018).

Risks

Our initial tests show that rendering each 3D model individually (each tile on the board being its own glb file) affects system performance, due to the size of the files. The lag generated includes screen tearing and slow moving scenes, however Babylon JS deals slightly better with large amounts of meshes in a scene compared to its competitors(Nordquist, O., & Karlsson, A. 2017).

We expect to be able to overcome this issue by limiting the amount of objects generated during a level.

Another risk when using BabylonJS is future proofing, due to BabylonJS getting relatively frequent updates, functions that worked previously may be depreciated. This has already occurred in our testing stages with a few examples, and this could be a problem down the line.

Despite this, due to the extremely active and helpful BabylonJS community, any depreciated functions will not be difficult to fix in the slightest.

Recommended solution summary

Babylon JS is a powerful web rendering engine, with plentiful documentation, and excellent performance. Sourcing 3D models from external vendors will save an enormous amount of time that could otherwise be spent improving Clara's 3D world code and performance. Finally, deploying the system to a University controlled server will allow full control over how the system is stored and deployed.

Alternative Solutions

Alternative Solution 1: Pixi JS

Solution Description

This solution aims at utilising the JavaScript library known as PixiJS as an alternative to BabylonJS to create a series of JavaScript files that will process and handle the files provided by the client to ultimately produce a three-dimensional view of Clara's world in which students will be able to interact with Clara through their inputted code.

Business Case

The reason as to why PixiJS was the chosen library is firstly because the library is free and opensource which means that the client does not have to worry about any licensing fees.

Furthermore, being that PixiJS was first established in 2013 and has only increased in popularity, this means that there is a vast wealth of knowledge accessible for free in terms of development.

Also, due to its popularity, many forums contain avid users of PixiJS which can help in periods of difficulty during the development period.

PixiJS is also very popular due to its speed, as it can quickly render objects due to its utilisation of WebGL and in instances where WebGL isn't supported, then the engine will revert to html canvas as standard. It is also touted as being multi-platform, meaning that users from either mobile or desktop will be able to take advantage of the benefits provided by PixiJS.

One example showing how to utilise PixiJS is the game YoHoHo.io (link: <https://poki.com/en/g/yohoho-io>)

It is important to note that BabylonJS is still the most popular engine for the game creation when compared to our alternative engines. As such, BabylonJS will have the most support and documentation available which will make it the more viable option during development.

Risks

In terms of risks involved utilising this development strategy for this solution.

- It would be on par with any other form of web development. It is more likely for developers to embed risks into the code during the development phase unknowingly, these forms of risks can be eliminated through thorough quality assurance.
- One potential risk that comes directly from the library itself is that an external module is needed to ensure that strict content security policy (CSP) is enabled, without this external module a door to cross site scripting (XSS) can be opened. This problem can be eliminated by using the aforementioned module ("PIXI requires unsafe CSP · Issue #7324 · pixijs/pixijs", 2022).

Alternative Solution 2: three JS

Solution Description

For alternative solution 2 we suggest using threeJS to create Clara's World in 3D. The main idea in using that solution will help the team to improve the workflow and help increase the efficiency of developing the project.

The best features that threeJS has is that it has many third-party plugins which supports the most demanding model formats. It's not hard to learn as there are plenty of good examples on how to use the features and on how to edit them and has a good API.

The Features that 3JS has is you can Export and import which helps to create JSON files easy and efficient, you can control the camera view at any point you like, has multiple shaders and objects which makes it great to make 3D Games and has good animation when trying to edit objects and scenes.

ThreeJS is a great software that is compatible with any available operating system and doesn't required an high end PC to run its features, which makes it great for any users to use without having to upgrade their hardware components.

Business Case

For the current interface for Clara 2D is that it is old and not user-friendly at all. The team is experiencing problems in which JavaScript program will be used to make Clara in 3D is that it won't be outdated and make it appealing for the user.

As a team we need to find the right solution so that we can identify the problem so that it can be fixed without having any issues.

For the solution, the team should identify which JavaScript program will be used and identify which program has better features, performance, rendering, API, easy to use and won't cost allot. So as a team we need to choose one solution from the three that will be the right choice to use to make Clara 3D.

Risks

For the Risk in solution 2 using 3JS is that

- Rendering is terrible to implement and really slow for beginners to use.
- Its features don't have allot stuff to use for rendering and scripting.
- The API is not that great and doesn't have advanced structures which are not show in the software.
- When it comes to making new application or models there's not allot of adjustment you can make because its main features are hidden.
- Extension for 3JS is not that good and needs improvement.

Alternative Solution 3: Developing Custom 3D Models

Solution Description

Benefits:

- Allows for better curated user experience, as each model would be individually made, the client vision could be better adhered to

Downsides:

- Enormous time investment would be required to make the models
- Enormous time investment would be required to learn the 3D modelling techniques required to make the models, as none of the team is familiar with/has any experience in 3D modelling

Business Case

More detailed models may provide a better user experience and would allow us to more closely adhere to the clients vision for the project.

Risks

The team has no previous experience in 3D modelling, hence a large initial investment of time in learning to 3D model would be required before any modelling actually takes place. There is no guarantee there is enough time allocated to the project to allow for such a large time investment, and while there is a chance for potentially high return on investment, there is an equal chance that the investment will return little to no reward.

Clara's World - High Level Business Functions

BFID	High Level Business Functions	Ranking
BF1	Generate Clara's 3D Board	Essential
BF2	Process User Interactions	Essential
BF3	Maintain System	Essential
BF4	Provide User Leaderboards	Non-Essential

Use Case List

An exhaustive list of all ways a user can interact with Clara's 3D World.

Function	Type
<ol style="list-style-type: none"> 1. Process user Interactions <ol style="list-style-type: none"> a. Process user code <ul style="list-style-type: none"> - Move forward - Turn left - Turn right - Invalid move (cannot move onto tree tile, water tile, off the board) - Grab leaf b. View User Leader-board <ul style="list-style-type: none"> - View high score achieved on each level c. Level Selection <ul style="list-style-type: none"> - Access is restricted to levels that have been progressively unlocked d. Graphics detail selection <ul style="list-style-type: none"> - Select high level graphics - Select medium level graphics - Select low level graphics e. Move camera <ul style="list-style-type: none"> - Swivel camera left - Swivel camera right - Zoom camera in - Zoom camera out - Increase level of declination - Decrease level of declination 	Essential

Development Release Schedule

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
BF1	Analysis	Design	Construction				Testing						Implementation	
BF2						Analysis	Design	Construction		Testing			Implementation	
BF3								Analysis	Design	Construction	Testing		Implementation	
BF4								Analysis	Design	Construction	Testing		Implementation	

Reference List

- *PIXI requires unsafe CSP · Issue #7324 · pixijs/pixijs*. GitHub. (2022). Retrieved 30 March 2022, from <https://github.com/pixijs/pixijs/issues/7324>.
- Nordquist, O., & Karlsson, A. (2017). *BabylonJS and Three.js - Comparing performance when it comes to rendering Voronoi height maps in 3D*. <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1228221&dswid=6774>
- Bawel, D. (2019, June 14). *Babylon.js vs. Three.js. . . Choosing a WebGL Framework for Sony*. HTML5 Game Devs Forum. <https://www.html5gamedevs.com/topic/37703-babylonjs-vs-threejs-choosing-a-webgl-framework-for-sony/>
- Bond, M. (2018, November 12). *Behind the Scenes with Adobe Dimension Engineers: How We Built the 3D Publish Feature*. Adobe Blog. <https://blog.adobe.com/en/2018/11/12/behind-the-scenes-with-adobe-dimension-engineers-how-we-built-the-3d-publish-feature>
- *Mercedes-EQ Formula E Speedboard Game*. (2021, November 29). Mercedes. <https://www.mercedes-benz.com/en/eq-formulae/we-drive-the-city/mercedes-eq-speedboard-game/>
- Pros and cons of three.js. (2022). Retrieved 30 March 2022, from <https://discourse.threejs.org/t/pros-and-cons-of-three-js/7050>