**The University of Newcastle**

**School of Information and Physical Sciences**



**Computing and Information Sciences**

**Work Integrated Learning**

**COMP3851A – Semester 1, 2024**

**Science and Engineering Challenge: Future Power**

*Author:*

Solomon Scobie, c3302821

# **Research Background**

The Science and Engineering Challenge (SEC) is a nationwide Science, Technology, Engineering and Mathematics (STEM) outreach program and part of the SMART Program (Science Maths and Real Technology) at the University of Newcastle (UON). Programs such as these aim to empower young people with a strong foundation in and inspire them to pursue STEM subjects in education. [1][2]

The SEC functions as an outreach program presented by the University of Newcastle, as a program within the university the SEC provides value to the larger organisation through positive word of mouth, encouraging students to pursue careers that the university can provide training for alongside providing educational value to students and teachers through events and professional development programs.

Community outreach programs are often well regarded by teachers, students and parents. Anecdotally having worked as a staff member at similar events targeted at children in Out of Home Care and Refugee Backgrounds; similar events have been very popular with students. Being fun days where they can engage with topics they would be unlikely to encounter in normal schooling and an exposure to novel environments and experiences.

The way the SEC accomplishes this is through a series of shows, workshops, Discovery Days targeted at year 5/6 primary school students and Challenge Days targeted at year 9/10 high school students.[3] Focusing on Discovery/Challenge Days, students are given gamified tasks, and asked to compete with problem solving skills. The Future Power task is one of these, and the goal of this WIL project is to transform that task from a physical game into a digital game to be hosted on a website. This seeks accessibility for harder to reach rural communities and or students who otherwise cannot participate in these in person events.

Australia has a long history of rural educational disadvantage.[4] While far from perfect initiatives like these may be able to give some students an interest in STEM subjects that they would otherwise be uninterested in pursuing.

Gamification of learning, outreach excursions, incursions and other techniques employed by the SEC have been shown to have positive outcomes in student interest and success in STEM fields.[1] Many governments and organisations focus on secondary educational programs to drive participation in STEM fields. A particular focus is often on Women in STEM due to their historical under representation.[2]

More technically the project is the development of a game in the game engine Unity. It is a demo based on the balancing of load and demand for a power grid, striving to avoid over or underloading a system and causing a loss of power, this serves as an entry level introduction to some concepts behind electricity and the “grid” we rely on day to day. In the presently existing demo students “demand” power from categories like “light industry”, “residential” or “heavy industry” and supply power from generators using a variety of sources renewable and otherwise.

Unity as a tool is a well-documented game engine [5] with a large community of creators creating tools, tutorials and paid assets for fellow creators to use and learn from. As a developer with experience in other game engines, working with the information provided by unity’s community has allowed for an easier pathway into working with the tools provided.

Other such educational programs and games exist, institutions like Questacon [2] and The Smithsonian [6] often run programs similar in goal to the SEC, and digital games are another potential tool in the arsenal of education. These games or programs are often simple but engaging problem-solving challenges directed at a particular topic. In this way, “Future Power” is intended to be the same as its many predecessors but with a unique spin on the topic of electricity and how we engineer our power grids.

# **Aims**

Prototyping – Functionally Complete

Generators

Provide varying amounts of power based on user selection.

Consumers

Consume set amounts of power and can be toggled on or off by user selection.

Overloading

When the system is above or below a threshold of demanded power to provided power, the system overloads and ceases functioning until fully reset.

Scoring

A number value based on the “cost” of the generators used to the “return” from power provided. Exact numbers are undecided and are to be discussed with supervisor further during the development.

User Interface – Functionally Complete

Deliberately Utilitarian for the prototype phase, currently designing and planning the final aesthetics of the UI. However, the UI allows a user to select what generators to use, how much to generate and what consumers are online. Provides information on what the current state of all assets are.

Backend Systems deemed unnecessary for the prototype – Awaiting Implementation

Scenarios

The physical demo is based on a series of “scenarios” that players must complete by providing power to pre-determined consumers in the most profitable way they can manage.

Score Recording and Reporting

Requires a functional Scenario system, complex score reporting is considered a “stretch goal” however simple score reporting once a student submits a solution to a scenario will be required.

Aesthetics – Currently in block out stage

Environment

A 3d room environment for the game to take place in with an altered design for the “game board” for a player to work with.

UI/UX

Translating the functionality of the prototype UI into the Environment the game is set in. Readability and Usability take priority, features like an instant reset button are currently missing in the prototype.

Sound

Considered a stretch goal to be completed once primary task is completed, Audio feedback for player actions, ambience, electric noises, to be discussed later with supervisor.

# **Methods**

A white box with knobs and a green screen

Description automatically generatedA wooden frame with black circles on it

Description automatically generatedInitial meetings were focused on determining the specifications of the original game and how it functioned. Access to the original was provided and after hands on testing notes on how the player controlled the game were taken.

Figure 1: The original game board, taken 28th March 2024

These meetings were however unable to provide full details on how aspects of the original game functional. Systems such as scoring calculations, the rate at which generators adjusted power output based on user input and the threshold for overloading are still unclear and actively being discussed with my supervisor.

Planning, notes on the current state of the project and notes on how to implement future features were recorded in an Obsidian Markup document that was committed along with all code changes to a GitHub shared with my supervisor.

Starting the project was primarily a focus on learning Unity’s User Interface tools, namely TextMeshPro. Due to inexperience with this particular engine this process took a few weeks to complete but now serves as a base for further development and improvement.

Creating generators, consumers and the systems involved therein was like most programming a simple process of taking numbers out of one box, doing an operation on them and then storing them in another box for later operation. Learning exactly how to communicate between different unity scripts and objects was slightly different to engines I have worked with, namely Unreal and S&box but did not pose a large challenge.

The whole game was designed around a “Game Board” object as a controller which held and communicated with objects below it, the game board controls scoring, total generation, consumption, overloading and other features. Generators, consumers and UI elements all deriving their data from either the Gameboard or the specific sub node a component was dealing with, to later pass that data onto the Game Board.

# **Results**

## ***Game Board***

As stated in the methods section the prototype architecture is based around a singular Game Board that controls all aspects of its particular version of the game. This does not reference any aspects of the scene that are not contained within itself and dynamically finds and sets up to work with components attached to it such as Generators and Consumers.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated This architecture is shown in Figure 2 with the Game Board receiving references to a Generator Controller and Consumer Controller node. From these Controllers the Game Board receives state data about each Controller’s sub nodes and calculates a score and whether we should change into the “Overloaded” state based on parameters that are yet to be fully decided for both occurrences.

Figure 2: Game Board Architecture

## ***Generators***

**A screenshot of a computer

Description automatically generated**Generator Nodes are themselves an individual Generator Node Controller with a child of a specific Node Type and a UI Canvas, the look of the prototype in game is shown in Figure 3. The default state for this Node type is None, an uninteractable Node that does not generate power. The user can select the dropdown menu from the Node Controller’s UI and change this node into any of the applicable Generator types.

Figure 3: Prototype Generator Node UI

The Controller receives an OnValueChanged event when the dropdown menu is interacted with and calls ChangeNode() deleting the old generator node and instantiating the selected Node Generator prefab; None, Coal, Gas, Nuclear, Hydroelectric, Solar or Wind. The scenario system is not implemented but will apply restrictions to which generators a user can or must use.

Each Generator has a generator capacity, response speed and target, this response speed causes the value of the generated power to linearly approach the targeted power with response speeds being relative to the type of generator, coal being the slowest and hydro being the fastest to mirror response speeds in the real electric grid. This process is frame rate independent and so should not differ on weaker or more powerful computers, taking a set amount of time to reach any particular value.

Each Node Controller queries the specific node assigned to it and passes the power generated onto the Generator Controller which totals all the incoming power before passing that information onto the Game Board. The relative amount of power each node generates is based on how far a slider bar is moved producing a proportion of the total possible energy a generator can output between 0 and 1.

## ***Consumers***

The Consumer Controller functions very similarly to the Generator Controller with the controller having a series of child “Consumer Groups”, each Consumer Group contains 1 to 6 consumers. Figure 4 shows the system with Two Domestic 1 consumers and a single Heavy Industry requesting Power.

The controller queries these groups, totals the requested power from each consumer and sends that through to the Game Board, when requested. Consumers are simple, having an on or an off state and a set amount of power they request based on their predefined type.

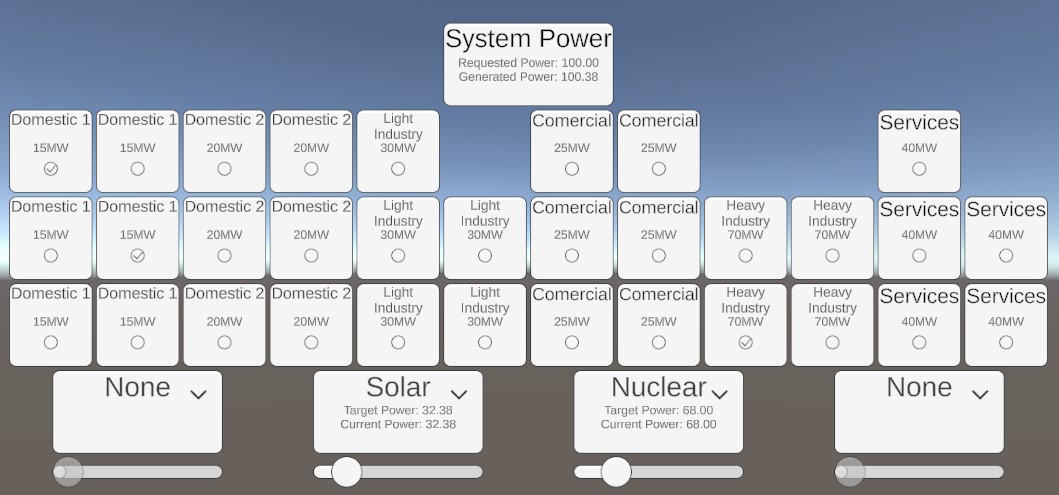


Figure 4: Prototype Consumer Node UI

## ***Scoring and Overloading***

Currently Scoring is functional, however, undisplayed in the UI. The scoring has had to be reverse engineered due to lacking access to documentation or source code of the original game board. Currently each generator has a cost to install, and a cost per mega watt generated, the real numbers of these things are hard to test as the board does not display a score unless you have successfully completed a scenario.

What is known is that the cost per mega watt becomes cheaper as a generator produces more power. The decision was made to move onto taking the project to a more complete state in the UI, Visuals and Functionality before returning to decide specific values for the scoring system.

Overloading is in much the same condition as scoring, overloading has currently been implemented with a “buffer” of 50 MW and a secondary buffer of 0.1\*GeneratedPower meaning that if the requested power is above or below the generated power by 50MW +- 10% of our total generated, the system will “overload”. It is known that this is not the functionality of the physical board, future in person testing will be arranged to reverse engineer a closer solution.

## ***Prototype UI and Future UI Drafts***

The prototype UI was deliberately designed to be utilitarian for speed of prototyping. It currently lacks, any aesthetics, hiding some raw numbers that are irrelevant to the end user, proper button signposting, tutorialisation for a new user and is all crammed onto one screen.

On going discussions and design meetings aim to include missing features, like score reporting, an instant reset and to work the UI diegetically into a 3d scene. This will require rounds of iteration and collaboration with Supervisors to achieve a readable visual experience. The current plan is to place a box into a 3d scene with a series of fixed camera angles a player can transition between, granting access to different parts of the game. e.g. consumers and generators having their own “screens” like in the physical demo. The specifics of this process are being designed and awaiting feedback.

# **Ethics**

The game has not encountered any real ethical issues so far. Ethics have limited scope of the project in a way that does not truthfully impact the development process. This game is not a paid experience and has no need to record personal or identifying information to function. As such, it will not do these things.

The potential of multi-player has been discussed as the original demo was intended for groups, and this may require handling the information of accounts or some similar system. In the case that this does become a part of the project, no data will be recorded once the program is shut down and of the data required to handle multiplayer it would likely only be in the form of a unique username to distinguish players.

It is a short problem-solving challenge, targeted at children and teenagers which does restrict the content of the game. The content of a simplified power grid simulation is not particularly explicit. Care will be taken with all language and content included however there is no need or desire for any form of explicit content.

The surrounding infrastructure of the project, and where it is hosted may have their own potential issues however this game is intended to embed within a larger webpage without interacting with it and as such, those concerns are out of the scope of this project.

# **Self-Learning and Project Management**

## ***Self-Learning***

Unity documentation and content from community members has provided avenues for understanding how:

the engine handles getting the children of a GameObject,

how to get only a specific type of GameObject,

how to work with TextMeshPro,

the different buttons, sliders and menu types provided as a base,

how to tweak those bases to fit your purpose,

how to reference UI elements in code through events or directly,

how unity’s C# differs from other engines,

etc

3d Art I have primarily worked with character modelling and animation blocking out a piecing together a scene to constrain the game within is in progress and not yet in a useable state but is an active process of learning

In the same way designing an “object” for the player to interact with is a process that very heavily influences the User Experience. Ongoing research is being done on products already existing and tutorials from other developers to achieve the best User Experience for this project.

Audio design has not begun but is a very heavily encouraged aspect of the system, I have little knowledge of the area personally and will require research on sound design software. However, the direction for the “soundscape” has been discussed and designed as being an unobtrusive ambience, some electrical sounds for the “box” a player will be working on and audio feedback noises such as clicks for when players press buttons, move sliders or otherwise interact with the game. I have the experience of implementing SFX and VFX in S&box and Unreal and hope to translate these skills into unity following a period of research into the ways that unity handles its sound systems.

The hope is that online creators will be able to provide a good starting point for creating these custom sounds when work begins on this system shortly

## ***Project Management***

Planning, notetaking and documentation has been done in obsidian. The initial designs of how the project would be structured, how features that became important during development would be implemented and the like are also initially designed on paper here before short informal meetings with my supervisor to discuss the desired direction.

Meetings have taken place more or less as required, with the longest gaps being 2 weeks with more frequent individual updates in between as progress is made. Larger meetings took place on the 26th of March, 28th of March and 2nd of May, these meetings discussed the overall direction and planted the base of the project, gave hands on experience and notetaking with the physical demo and discussed the audio-visual direction of the final product.

These individual meetings have kept the goal clear and both parties aware of the current state of the project. The prototype phase is considered complete and moving forward will be the development and iteration of builds intended to be more final rather than a functional proof of concept.

# **References**

[1] Different kinds of STEM education initiatives, Department of Education, <https://www.education.gov.au/australian-curriculum/national-stem-education-resources-toolkit/i-want-know-about-stem-education/different-kinds-stem-education-initiatives>

[2] Action area: Enabling STEM potential through education, Department of Industry, Science and Resources, <https://www.industry.gov.au/publications/advancing-women-stem-strategy/vision-equal-opportunity-stem/action-area-enabling-stem-potential-through-education>

[3] What is the Science and Engineering Challenge?, Science and Engineering Challenge, <https://www.newcastle.edu.au/college/engineering-science-environment/education/science-and-engineering-challenge>

[4] Schooling and higher education in rural areas, National Rural Health Alliance, <https://www.ruralhealth.org.au/news/schooling-and-higher-education-rural-areas>

[5] Unity Website, <https://www.unity.com>

[6] Educational STEM Games for Students, Smithsonian Science Education Center, <https://ssec.si.edu/stemvisions-blog/educational-stem-games-students>