The STEM Game

The Problem

Thales is an Engineering company and requires a constant stream of talented people to join them each year during their recruitment. Part of the company's commitment to investing in young people is by getting involved in activities at schools and universities to promote STEM subjects (Science, Technology, Engineering and Maths).

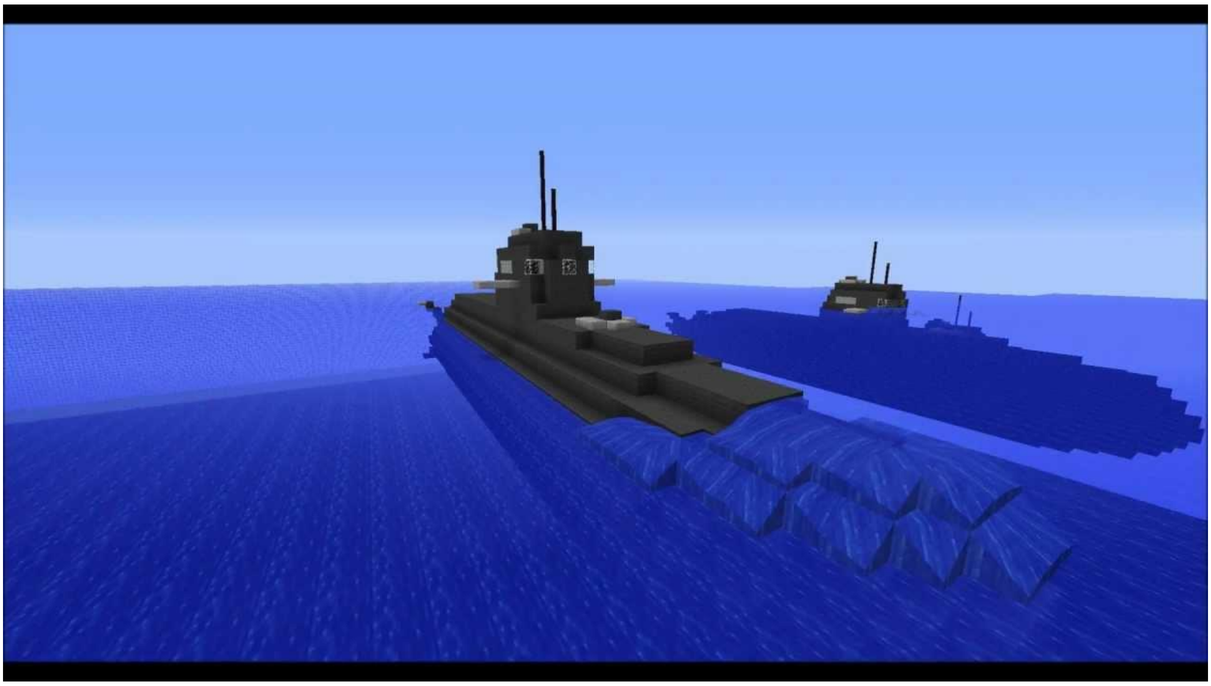
In order to support these events and especially with the younger students, the need for innovative solutions to engage and inspire the Engineers of tomorrow is essential.

We require an interactive game to showcase our business and innovative products to young people in the age range 10-15.

The Solution

Minecraft is a popular game with people of various ages, especially children. Creating a world in Minecraft can be achieved on a Raspberry Pi using the Python programming language. This enables a developer to create blocks, interactions and complete worlds quite easily using the Python API.

You are required to create "Thales Worlds" using Minecraft Pi. The world should have a Thales theme and should showcase Thales products and the work which we do.



Research topics should include:

- The sectors that Thales operates in.
- Thales products.
- How to use the Python API to manipulate the Minecraft engine.

You should pay close attention to the age range of the target audience to ensure that the game is interactive and fun for people in that range.

Project Milestone / Feedback

There is an opportunity to showcase the product at a STEM event in March, therefore we would like to have a working milestone delivery for the middle of February, it does not have to be a completed version of the product but must be usable.

The STEM event can provide you with essential end user feedback which can feed directly back into your final development phase.

Deliverables

A working Minecraft world as well as the following things:

- Project report detailing the following:
  - Project plan – detailing how the project will be executed, consider timescales (Gantt chart, critical path).
  - World design – Storyboards, level design, block design etc.
  - Lifecycle choice – Agile? Waterfall? Assess different methods and provide reasoning for your choice.
  - Technologies – do you need additional technologies to achieve the goal?
  - Application testing – how you test your application, unit testing, regression testing and formal testing.
  - Evaluation – evaluate your product against the requirements, highlight areas that you would improve or perhaps things you would have done differently if you had more time.
- A user manual which outlines how to install/run the product.