



MINECRAFT

EDUCATION EDITION

Educator Guide

Seymour Island

2-3 Hours

Single Student or up to 29 Students (plus a teacher) in Multiplayer

Seymour Island

INTRODUCTION

Seymour Island is a multiplayer supported, “open world” style adventure, featuring a series of 8 coding tasks for students to complete.

The world was designed to be completed with the full class of up to 29 students (plus 1 teacher), playing within the same shared world. Throughout, students earn points for each task they complete, which allows for progress to be tracked.

Although the world can be completed by a single student, it is highly recommended that the world is played with 2 or more students, as task 6-8 require a second player to help.

As this is an “open world” style activity, beyond the initial tutorial section (task 0), students are free to complete the available tasks in any order they wish to.

The world makes use of the MakeCode platform for coding. Each task (with the exception of task 0) is available both in MakeCode Blocks and MakeCode Python. This gives students the option to try out and explore either language, depending on their preference or coding level.

Finally, it is advised to allocate students a dedicated “partner” or “buddy”. This is due to the fact that tasks 6-8 require an additional student to complete.

STUDENT PREREQUISITES

The following is assumed of students attempting the world

- They are able to move around the game, including using right/left click.
- It is recommended that the Hour of Code introduction world has been completed prior to attempting Seymour Island.
- Although the world can be completed by a single student, it is highly recommended that the world is played with a minimum of 2 students, as task 6-8 require a second player to help.

THINGS TO KEEP IN MIND

- In each task, students are given a Blaze Rod, which they use to execute the code they have written. This is instead of the usual Play button.
- Remind students there may be more than one solution for each of the activities.



START OF LESSON PROCEDURE

LESSON REVIEW: 5 minutes

Recap with students on movement/controls, along with what they learnt from the Hour of Code lesson. Ask the students:

1. Q. What were 4 keys on the keyboard that move you around in Minecraft?
A. WASD
2. Q. How can students access the Code Builder window?
A. Press the C key on the keyboard.
3. Q. How can students close the Code Builder window?
A. Press the ESC key on the keyboard.

MULTIPLAYER GROUND RULES: 10 minutes

With this world being a multiplayer world, where potentially the entire class will be playing together, it is important to set some ground rules for the students. Some suggestions can be found below.

- Do not interfere with anyone else's work or activity - As they will all be playing together; it is possible for students to try to get in the way of other students or interfere with their work. This makes the process less enjoyable for everyone and wastes both of their time.
- Engage in the tasks available – With a world this large, it is common for students to hang around in the courtyard, the town hall or down in the mines, avoiding completing the tasks. This once again wastes that students own time. They are being given the freedom in this world to explore and make their own adventure, if they abuse this then they will no longer be able to take part in the activity.
- If a student needs help, ask their partner first – This helps reduce the workload on the teacher and encourages students to collaborate with their “buddy” or partner, with that student operating as a first level triage for issues or problems. Only if they are both unable to solve a particular issue, then they should come to the teacher.
- Something not working? Have they double checked the instructions? – In many cases, students have struggled with some of the more complex (red level) tasks in the world simply due to then not reading the instructions for the task. Encourage students to read the instructions and plan their coding solution, before sprinting straight into coding.



LESSON INTRODUCTION AND LEAD-IN: 5 minutes

Lead-in:

Tell the students they will be “heading on an epic adventure across the seas”, but it isn’t all going to be plain sailing... They will have to work together with their partner/buddy to solve a series of tasks to help out the residents of Seymour Island.

These tasks will be labelled with the following colours to signify difficulty

- **Green** – Easy/Beginner activities.
- **Orange** – Medium/Intermediate activities.
- **Red** – Hard/Advanced activities.

Although once students complete the introduction, they can choose their own adventure pathway through the tasks, make the strong suggestion that they should start with the beginner activities (green), before working their way up to orange and finally red.

STARTING OFF THE WORLD: 10 minutes

As students join into the world, they will find themselves in the Lobby.

The Lobby contains a number of jumping challenges to keep the students occupied until it is time to start the activity.



Figure 1 - Student Lobby



As the students join, it is advised to inform them of their “buddy” / “partner”. Although they will not need this as part of the first task (task 0), they will require knowing who their partner is by the time they reach the main island later on.

TEACHER CONTROL BOX



Figure 2 - Teacher Control Box

Directly above the Lobby is the teacher control box. This includes a button to start the game, along with a button to teleport the teacher to the main control room (in the town).

The teacher can simply fly up, by changing their gamemode (using the /gamemode c command in the game chat) or by teleporting there using the following command in game chat /tp @p -871 153 379

When you are ready to start off the activity, right click on the button below the activity start sign (on the right in the above image). This will allocate each student currently in the game, out to an individual boat.

If any students arrive late, it is possible to press the button again and they will also be allocated out onto their own individual boat.



TASK OVERVIEW

Below is a list of the tasks available to the students.

[A full list of solutions for each task can be found here.](#)

TASK 0.1 – MOVING AGENT

Task 0 is a 3-step process in which students are introduced to the basics of the Agent. This takes place for each student on their own boat, guided by Captain Pappert.

In task 0.1, the student must locate their agent, then move it 2 blocks to the left.

If students have any issues, they should right click the button on the left hand side of the boat, on the wall marked “Reset Full Task”

TASK 0.2 – BREAKING A BLOCK

In task 0.2, the student must use the destroy forward block to break the orange wood block in front of the agent.

TASK 0.3 – REPAIRING THE HULL

In task 0.3, the student must move the agent to the left by 3 blocks, then place a block to its left. This repairs the hull and the player is informed that the task is complete and they should navigate to the captains quarters to speak with the captain.

After doing so, they are instructed to “grab the steering wheel!”. Once students do this, they are shipwrecked.

TASK 1 – FISH AND MELON

The Fish and Mellon is a beginner task located in the main town square. It involves students using their agent to place down 3 pieces of redstone to complete a circuit.

This activity is generally completed by students in 3 sections, as opposed to a full large simple program.



TASK 2 – AUTOMINER LEVEL 1

The autominer tasks are a series of 3 tasks that offer a progression pathway up the levels.

In Autominer 1, students are asked to dig a 7x1x3 long tunnel using the Agent. This area is behind glass which students are unable to break, so it must be done by the agent.

This task encourages students to use a Loop (For Loops) approach but can be completed by a series of smaller programs.

TASK 3 – AUTOMINER LEVEL 2

In Autominer 2, the task progresses from the previous with the students being required to this time, only mine the Iron Ore. Mining any other block will result in the task not being able to be completed. This requires the students to use an If Statement, as well as Loop (For Loops). This can be completed instead though with a series of smaller programs.

TASK 4 – AUTOMINER LEVEL 3

Autominer 3 adds 2 additional points of complexity beyond Autominer 2. It adds the requirement for students to mine any one of a number of different ores in the wall. This requires them using If Statements and Else Statements.

In addition, there is also a 50 second timer applied for this task once the agent moves off the starting position.

If the program does not successfully mine all the required ores within 50 seconds, the task (although not the students actual code) is reset.

TASK 5 – FISH UNLOADING

In the Fish unloading task, students are required to use their Agent to unload a total of 50 Clownfish and a total of 50 Pufferfish from the ship.

Two collection hoppers can be found on the pier, where the fish should be brought to and placed in the correct hopper.

This task encourages the use of loops (While Loops).

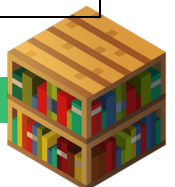
TASK 6 – SHOCKING DISCOVERIES

Shocking Discoveries is a multiplayer task, requiring 2 players. The task is a much larger and more complex version of Task 1.

It features 2 circuit areas for each student, which must have redstone placed to complete the circuit.

In this task, the 2 students have a very similar task to solve to automatically place the redstone, so they are encouraged to help each other.

This task encourages the use of If Statements and Loops (For Loops).



TASK 7 – BEGINNER MULTIPLAYER MAZE

The beginner multiplayer maze requires 2 players to complete. It involves one player being the “Driver” (they write the code) and the other being the “Navigator” (they help guide the Driver through the maze).

The navigator is able to see the maze that must be solved but is unable to program an agent. On the other side, the Driver is able to write the code, but can’t actually see the maze. Whereas, the driver is able to write code for their agent but is unable to see any of the invisible maze.

Note this task requires students to be ideally on a voice call (or right beside) their partner.

TASK 8 – INTERMEDIATE MULTIPLAYER MAZE

The Intermediate multiplayer maze requires 2 players to complete. It involves one player being the “Driver” (they write the code) and the other being the “Navigator” (they help guide the Driver through the maze).

The navigator is able to see the maze that must be solved but is unable to program an agent. On the other side, the Driver is able to write the code, but can’t actually see the maze. Whereas, the driver is able to write code for their agent but is unable to see any of the invisible maze.

This maze contains a series of pistons and pressure pads, which must be triggered to allow the agent to pass.

Note this task requires students to be ideally on a voice call (or right beside) their partner.



LESSON CONCLUSION: 5 minutes

At the end of each lesson, you may want to discuss the following questions with students.

1. Q. Which activities did you find the most interesting? Why?
2. Q. Did you find working with your partner to be useful? What sort of approaches could you use in the next lesson to work better or more efficiently with them?

COMMON STUDENT ISSUES

There are a number of common issues students have in Seymour Island. These can be found below

1. **Right clicking with a blaze rod results in nothing happening, even though student code is running.**

This is a known bug within Minecraft Education Edition itself. In general, the best way to resolve it is ask the student to restart their game entirely (not just log out of the world and back in).

2. **In the autominer tasks, a student asks why they haven't completed the task, yet their agent is standing on the final pressure pad.**

This is most commonly because they either haven't actually removed all the required blocks (mainly autominer 1), or have removed too many of the stone blocks that they shouldn't have (autominer 2/3).

3. **A student's agent is missing or isn't teleporting to the right location.**

This is a known bug within Minecraft Education Edition itself. In general, the best way to resolve this is to ask the student to restart their game entirely (not just log out of the world and back in).

4. **In the initial shipwreck section of the world, students are unsure of where to go after speaking with the guard.**

Encourage the students to explore, reading the clues within the world on signs. They will have to backtrack on the bridge, then climb down the ladder to the side. From there, they should be able to see a conveniently located sewer with smoke emerging.

5. **There are leftover scoreboard entries on the right. How can these be reset?**

To reset all scores for students, type the following command into the game chat
`/function reset_scores`

6. **The wrong student or multiple students have ended up inside the same task.**

This can occur due to selection limitations within Minecraft Education Edition. Encourage students to quit the task using the buttons on the wall and try and join again.



7. **The codebuilder opens, but all is shown is a running agent character.**

This is a known bug within Minecraft Education Edition itself. In general, the best way to resolve this is to ask the student to restart their game entirely (not just log out of the world and back in).

KEY TEACHER COMMANDS

Minecraft standard commands

In game command	Purpose
/tp STUDENT-NAME	Teleport to student
/tp STUDENT-NAME YOUR-NAME	Teleport student to you
/clear @a	Clear all student inventories
/give STUDENT-NAME blaze_rod	Give a student a blaze rod (if they lose theirs)
/gamemode c	Set yourself to creative mode (so can fly)

Seymour Island specific commands

In game command	Purpose
/function lobby	Teleport to the student lobby
/function control	Teleport to the main control room
/function settings	Enable all default game settings
/function reset_scores	Reset all student scores



MAPS

Student Map



World overview map

