



RENT-A-TECH

 60-90 minutes

 Grade 5 and up

 Team

 Intermediate

Micro:bit Hot Potato

Hot Potato is a timeless game played by multiple people of all ages. The group stands together and passes around something that resembles a potato (usually a beanbag), while singing or playing a song. When the song ends or the music stops, the person holding the potato is eliminated. The game goes on until just one player remains. In micro:bit Hot Potato, you will use a micro:bit as the potato and radio frequency instead of a toss.

Goals

1. Practice the computational thinking aspect of decomposition to understand the steps required to code the activity.
2. Become familiar with using multiple blocks in a sequence.
3. Have fun playing as a result of the work.

Skill development

- Collaboration
- Goal-oriented
- Critical thinking

Prerequisites

- The facilitator should have a basic knowledge of micro:bits.
- Participants should have completed some introductory activities with MakeCode software.

Digital Equipment

Each participant requires:

- Micro:bit
- USB cable
- Battery Pack
- Batteries
- Computer

Curriculum links

All grades

A1. Apply to the best of their ability, a variety of social-emotional learning skills to support their use of the mathematical process and their learning in connection with the expectations in the other five stands of the mathematics curriculum.

Grade 5

C3.1 Solve problems and create computational representations of mathematical situations by writing and executing code

C3.2 Read and alter existing code, and describe how changes to the code affect the outcomes.

Grade 6

C3.1 Solve problems and create computational representations of mathematical situations by writing and executing efficient code.

C3.2 Read and alter existing code that involves events influenced by a defined count and/or sub-program and other control structures, and describe how changes to the code affect the outcomes and the efficiency of the code.

Coding Concepts

This activity relies primarily on the computational thinking aspect of decomposition, which requires participants to understand how the game is played in order to create the activity steps. This tutorial breaks the activity into five coding steps.

In the table below, the left column represents how a practical game of hot potato is played. On the right are the decomposition steps, broken into codeable actions for the MakeCode software. They follow a similar sequence to the real game, but using software language.

	Practical Game	Decomposed actions
1	Pick an object to represent the potato.	Create an object in code that will represent the potato.
2	Assemble a few players.	Define how to pass the potato between participants, using a radio signal set to the correct frequency. Players use the same radio frequency on their own code to interact.
3	Start the music.	Set a randomized timer. Define the players' action:
4	Toss the potato between players as quickly as possible.	<ul style="list-style-type: none"> • If you have the potato, send it to the other player. • Catch the potato when your opponent shakes their device. • Randomly determine how often the potato transits between players.
5	Determine who is the winner and loser based on who was holding the potato when the music stopped.	Define what the potato is doing: <ul style="list-style-type: none"> • If the game is done, show the losing screen. • If the player does not have the potato, show the blank screen. • If the player has the potato, start the timer count down.
6	Play	Save progress Add code to micro:bits Play

Preparation

Load the tutorial: **Hot Potato**

- https://makecode.microbit.org/beta#tutorial:github:exploratekcat/rent-a-tech_tech-a-louer/Hot_Potato

Confirm that Makecode is set to the participants' language. Click the gear logo on the top right, and choose the language tab to modify it (if required).

Facilitation

Present the activity

The introduction is necessary to create enthusiasm in the learners and get them in a creative and problem-solving mood. It will also ensure all the students starts from the same base, as some may be familiar with the activity while others will have no context for it. Some ideas for openers you can include in your presentation:

- *Has anyone ever played Hot Potato?*
- *What is the goal of this game?*
- *How do you play; what are the rules?*
 - If possible, have them play the game to experience the rules in context.
 - You can also make this question a short unplugged activity, individually or in a group, that introduces them to the decomposition of steps.
- *Let's try to code that game, and send each other virtual potatoes!*
- *How would we go about coding that?*
 - *Reference the steps they came up with during a practical game.*

Before starting

1. Assign students to teams of two.
2. Give each team a radio channel number. Each member of the team must use this specific identifier in their code.
3. Distribute one micro:bit, one USB cable, and one battery holder (with battery) to each student.



Safety notes

The micro:bit is an exposed circuit board. This allows you to see all the electrical parts and understand how computers work. This does pose a small risk of parts being damaged, so remind students to handle with care.

During the activity

1. Instruct participants to plug in the micro:bit to their computer, using the USB cable.
2. Ask participants to open the **Hot Potato** tutorial, using the URL provided.
3. Each participant must write their own code for their device.
4. Ask them to follow along the tutorial steps.
 - a. By clicking on the clue tab, an image of the intended block will appear.
5. Once their code is completed, they must save it to the computer.
 - a. Define a location for participants to save their files.
6. Transfer the saved .hex file to their micro:bit.



RENT-A-TECH

7. They can use the micro:bit while it's plugged in, or replace the USB with the battery pack.
8. Ask participants to test their code; can they send the potato?

Some questions that students might have:

1. Why isn't my micro:bit working?
 - a. Make sure the code is saved and downloaded to the micro:bit. It needs to be updated every time you make a change to the code.
2. Why doesn't it work with my partner?
 - a. Ensure that you are both using the same radio frequency.

Assessment

The questions below can be brought up as a group discussion, or written down on paper.

- What was the most challenging part of the activity?
- Were you able to follow the tutorial on your own, or did you require extra support?
 - What did you do to find answers?
 - Were you able to troubleshoot on your own?
- Did you and your partner work together well?
- Did you try modifying the code a little to see what would happen?

Go beyond

Now that they have the base coding done, participants can perform modifications to implement easy or more challenging changes. Here are some ideas to inspire them:

1. Can you display something other than a potato?
2. Can they make the game longer or shorter?
3. Can you add more than two players to your game?
4. Can you code your micro:bits so the whole class can participate in the same game?
5. Could you add a sound to each potato pass, using the headphone and pin functions?

Tinker with this activity

Since part of this activity is to shake the micro:bit, ask participants to create secure packaging in the shape of their choice using a variety of recycled materials. The display and buttons must remain accessible.

Appendix

Overview of tutorial block functions

- **Basic functions:** These consist of pieces of code preprogrammed to do specific actions, such as launching programming, displaying on the micro:bit screen, or repeating the code that will be inserted.
- **Radio functions:** Data are sent from one micro:bit to another using radio waves. Specific blocks will allow you to create distinctive channels for communicating between devices, or send specific types of data.
- **Logic function:** These blocks account for different possible outcomes, and program their next action.
- **Variable function:** This is what the micro:bit software remembers and refers to, when requested. It can be a text, known as a string, or a series of functions. The first step to using a variable is to define what it is.
- **Input functions:** Each micro:bit device has buttons and integrated sensors that detect movement, light, temperature, and magnetic field; all of these can be activated through the various blocks of input functions.

Terminology	Definition
Function	Function refers to a category of blocks that are pre-programmed to perform one or many specific tasks.
Block	A block is a command container that holds ready-to-use code. Once added to other blocks, like a puzzle, the blocks create the program. If learners click on Javascript, they will see the long code written for each block.
Radio	Radio refers to the radio emitter embedded in the micro:bit device.
Variable	A variable is an “object” coded by the user in the micro:bit program.
Sequence	A sequence is a series of programming blocks.
Decomposition	Decomposition is an aspect of computational thinking, which consists of the breakdown of a complex problem into smaller tasks.