Tele-Potato

Don't keep the potato too long or you might lose the game!


Do you know the "Hot Potato" game? You toss around a potato while a timer counts down and the person holding it when the timer is up loses… It's super fun.

In this project, we'll build a similar kind of game but instead we'll use a **virtual potato** and the micro:bit radio.

Teleporting potato?

Instead of passing a real potato around while a real clock counts down, we are going to **send a number** between micro:bits. We can do that using the `Radio` blocks. They use the antenna on the micro:bit to send data over radio frequency signals, just like the phones or gadgets around you.
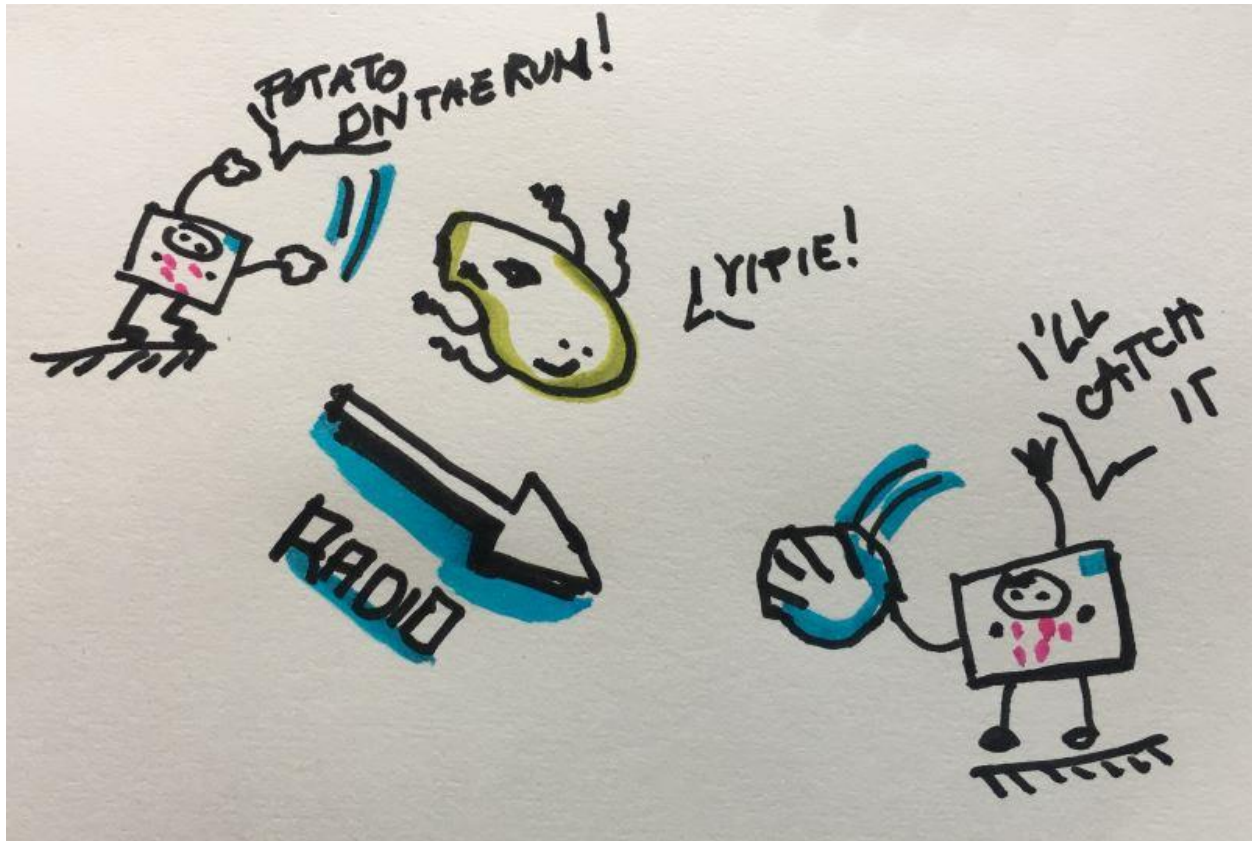
Now, what does it mean to have a number represent a potato? Well, we need to **model** the clock as **a number** being tossed around with the potato. We do this so that we can play the game using the radio. So what is so special about this potato clock? It ticks down the time and when it reaches 0, it rings.

To keep track of things, let's have a variable called **potato**:

- If the value of **potato** is positive, the player has the potato and the **potato** variable represents the **remaining time**
- if value of **potato** reaches 0, the game is over
- if the value of **potato** is negative, this means that the player doesn't have the potato in their hand
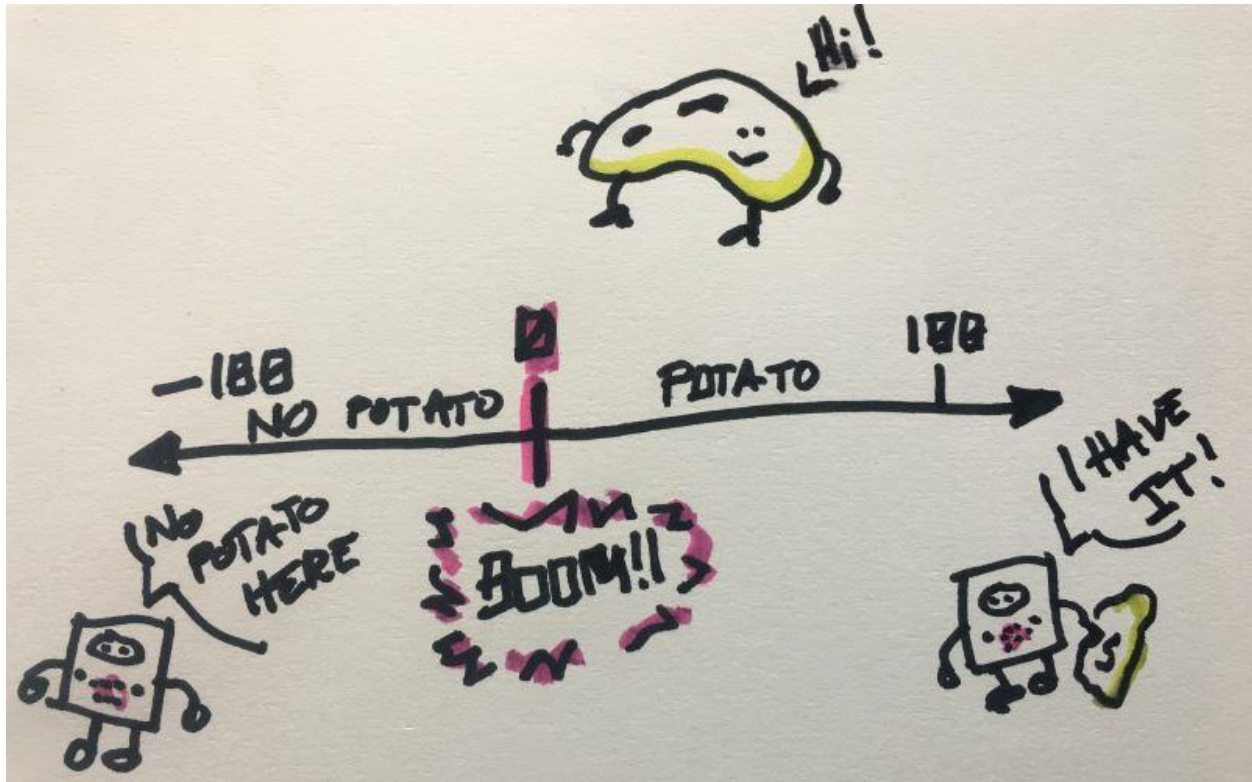
Now that we know what the potato is, we need to come up with the user interactions. This is how will the user play the game:

- press the `A+B` button to start the game and send the first potato
- when a potato is received, the screen displays some image
- when the player shakes the micro:bit, they send the potato to other players

Let's get coding!

Initialization

Let's start by creating the potato variable and initializing it to `-1` in `on start`. Remember, a negative potato value means you **don't** have the potato. We use `radio set group` to make sure players receive the messages.
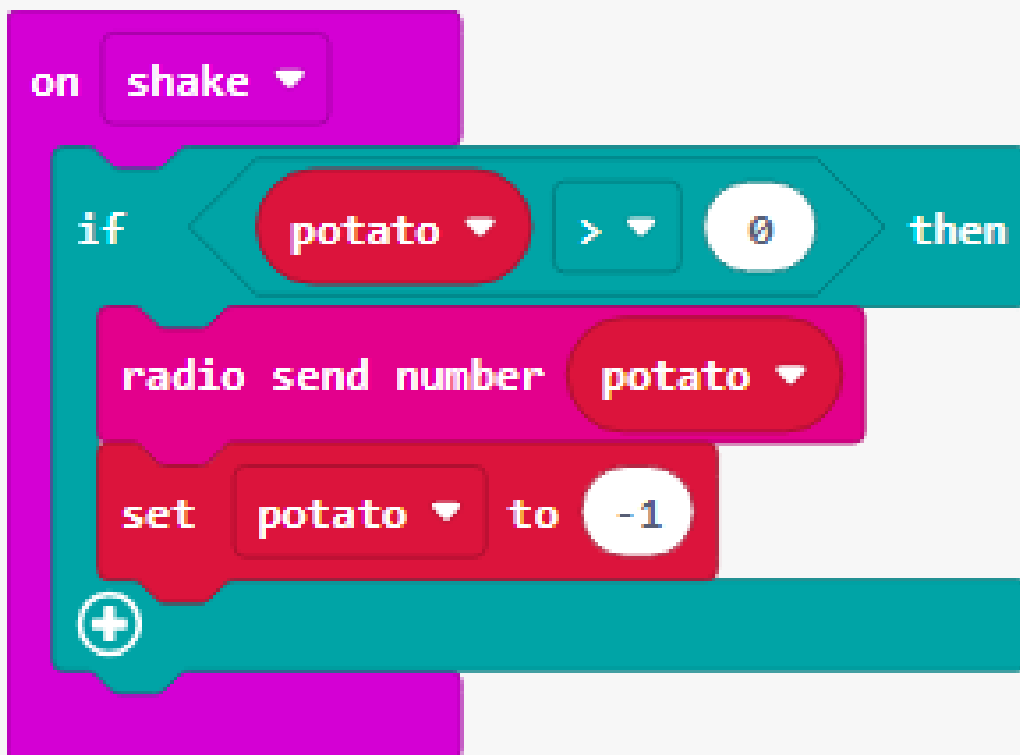
Starting the game

To start the game, we respond to the `A+B` button press and assign a positive number to the **potato** variable. To make the game less predictable, we use the `pick random` block to generate a value between `10` and `20`.

```blocks
on button [A+B ▼] pressed
    set [potato ▼] to (pick random (10) to (20))
```

Sending the potato

Sending the potato is done by shaking the micro:bit. If the **potato** variable is positive, we have the potato and we can send it. After sending it, we set the **potato** variable to `-1` since we don't have it anymore.

```blocks
on [shake ▼]
    if ([potato ▼] [> ▼] (0)) then
        radio send number ([potato ▼])
        set [potato ▼] to (-1)
```

Receiving the potato

Receiving the potato is done in the `on received number` block.
The **receivedNumber** represents the potato and is stored in the **potato** variable.

Ticking the clock

Making the clock tick down is done with a `forever` loop.

- If the **potato** is equal to `0` (`potato == 0`), KABOOM! you lose!
- If the **potato** variable is negative (`potato < 0`), we don't have the potato so we clear the screen.
- If the **potato** variable is positive (`potato > 0`), we display a potato image and decrease the variable by `1`.
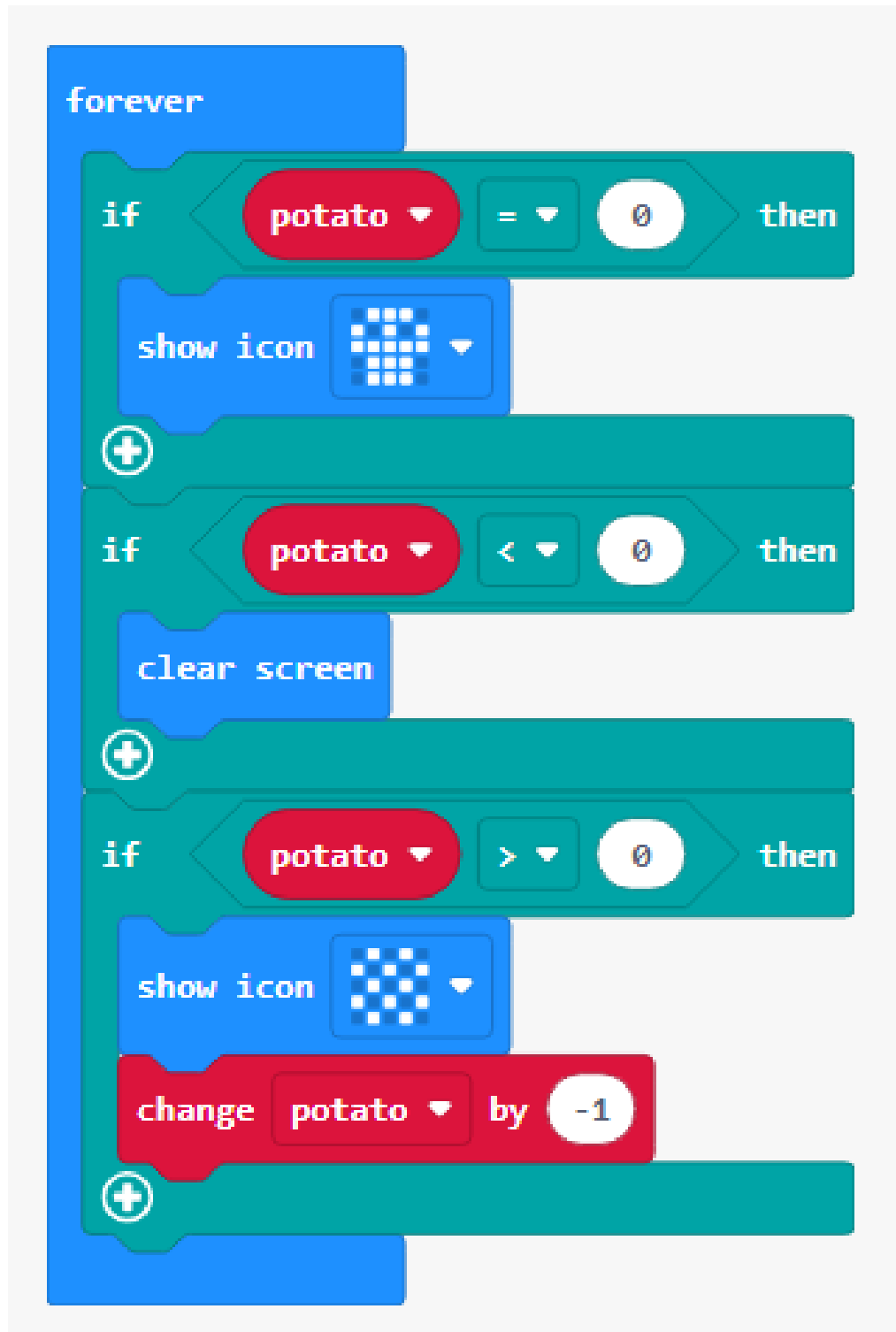
It's game time!

Get two or more micro:bits and download the code to them. One player starts the game by pressing `A+B` to receive the first potato. Good luck!