

Ready, Set, React! Build a PicoBricks Reaction Game

Picobricks

Beginner

30 Mins

1 Overview

In this project we are going to design a game to test your reaction times against your friends.

When PicoBricks is turned on, a timer will start. The timer can detect time in milliseconds, which are tiny slices of time. Timers are used in lots of household items like ovens, air fryers and microwaves.

We will use the OLED screen to show a welcome message when PicoBricks starts and instructions on how to begin the game. The timer will count down from three. Once it hits zero, the LED will turn on randomly anywhere from 1 to 5 seconds later.

When the LED turns on, the player has to press the button as fast as they can. This is the player's reaction time, and it will be measured in milliseconds. The player's reaction time will then be displayed on the OLED screen.



What you'll learn

- How to make a countdown timer
- How to use random numbers to surprise the player
- How to measure time in milliseconds
- How to make a button, an LED and a screen work together

2 What you'll need



What you'll need

- PicoBricks Kit
- PicoBricks Brick IDE
- MicroUSB to USB cable
- Laptop or tablet for coding

3 Key words

Key words

Millisecond

A millisecond is a tiny slice of time. There are 1,000 of them in just one second! That's why it's perfect for measuring things that happen really fast.

A blink of an eye takes around 300 milliseconds, so a really quick button press might be even faster than a blink.

Integer

An integer is just a whole number, like 1, 2 or 3. It has no bits after a dot, so 4 is an integer, but 4.5 isn't.

Counting down 3... 2... 1... uses integers, but a price like £4.99 isn't an integer because it has numbers after the dot.

Random

Random means you can't guess what's coming next, a bit like rolling a dice. The computer picks for you, so you get a different surprise each time.

Asking a computer for a random number between 1 and 6 is just like rolling a dice, you never know if you'll get a 1 or a 6.

Function

A function is a chunk of code you give a name to, so you can use it again and again without building it from scratch each time. Think of it like a favourite recipe, you write down the steps once, then follow it whenever you need it.

Like writing out a pancake recipe once. Whenever you fancy pancakes, you just follow the recipe again rather than working it all out from scratch.

Variable

A variable is something that can change. Think of it like a labelled box: you pop something inside, and the label tells you what it is. You can swap what's inside whenever you like.

A variable called score might start at 0, then change to 1, 2 and 3 as you collect points in a game.

While loop

A while loop keeps repeating its instructions for as long as something stays true. The moment that thing stops being true, the loop stops.

"While it's still raining, keep the umbrella up." As soon as the rain stops, you put the umbrella down, and the loop ends.

Forever loop

A forever loop repeats its instructions again and again without ever stopping (until you switch off or reset). It's perfect for anything you want to keep happening over and over.

Like a clock's second hand that keeps ticking round and round, a forever loop keeps running the same steps over and over.

Button

A button is a switch you press to tell your project to do something. The code can check whether it's being pressed or not.

Pressing a doorbell is just like pressing a button, one press sends a signal that makes something happen.

LED

An LED is a tiny light you can switch on and off (and sometimes change colour) with your code. LED is short for "light-emitting diode".

The little standby light on a TV is an LED, your code can flash one on and off the same way.

OLED screen

An OLED screen is a small display that shows words, numbers and pictures from your code. On some kits it's labelled the "LED display", but it's the same little screen.

Like the small screen on a fitness watch showing your steps, an OLED screen can show a score or a countdown.

Buzzer

A buzzer is a tiny speaker that makes beeps and tones when you send it electricity. By telling it different numbers, you can make it play notes and tunes.

The beep a microwave makes when your food is ready comes from a buzzer.

PicoBricks

PicoBricks is an all-in-one coding kit that joins up handy parts, like a screen, a buzzer, buttons and lights, onto one board, so you can build projects without any fiddly wiring.

It's a bit like a Lego baseboard for electronics: everything clicks together in one place, ready to code.

Raspberry Pi Pico

The Raspberry Pi Pico is a tiny, low-cost computer chip on a small board. It's the "brain" that stores your code and tells the other parts what to do.

It's like the brain inside a toy robot, small enough to fit in your hand, but clever enough to run your whole project.

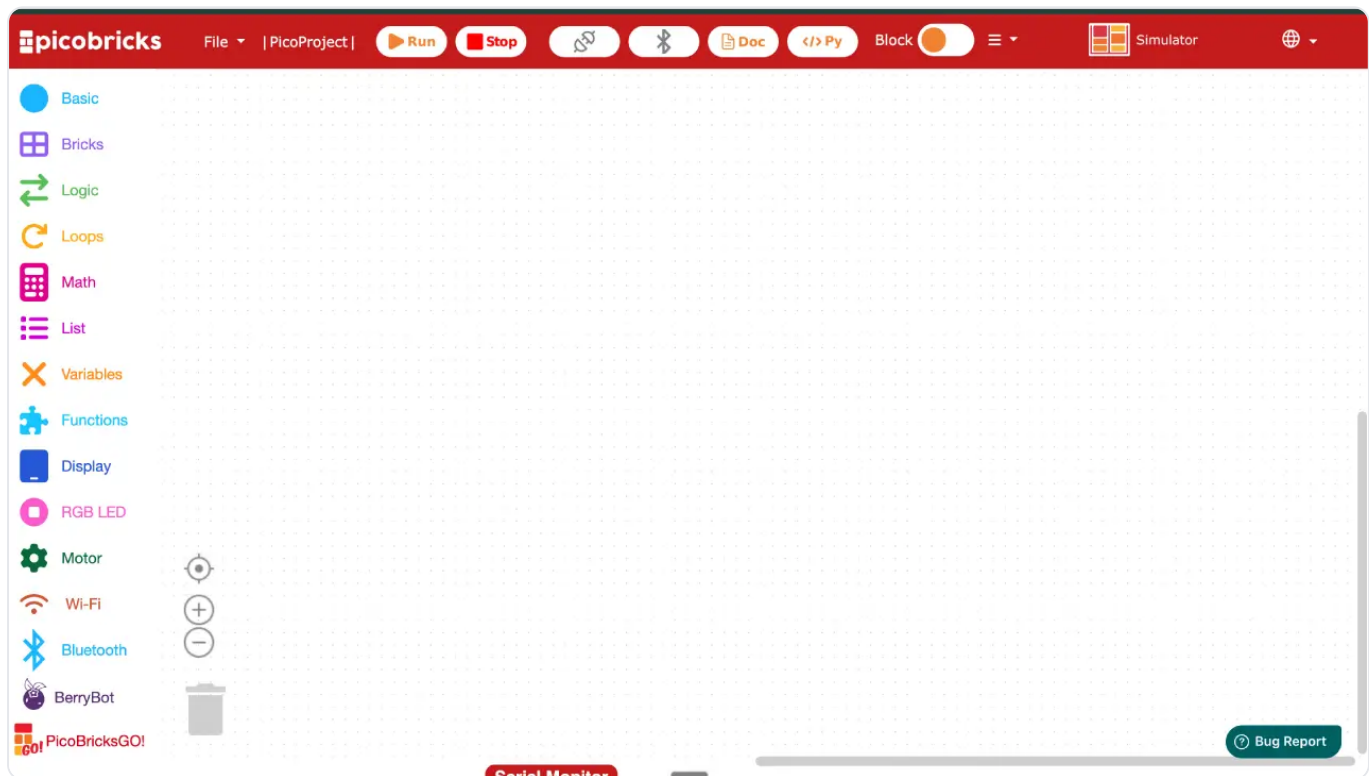
Module

A module is a small part that does one job, like a buzzer, a button or a screen. You connect modules to the main board to give your project new powers.

Like adding different attachments to a toy, each module snaps on to add a new ability, sound, light or buttons.

4 Set up your PicoBricks

1. Open your favourite web browser. We recommend Google Chrome or Microsoft Edge.
2. Type ide.picobricks.com into the address bar to open the PicoBricks coding editor.
3. Connect the PicoBricks board to your computer with the USB cable.
4. Select the **Connection** button at the top of your code editor to pair the Pico with the code editor.



5 Create the Countdown function

First, we will create a function called `CountDown`.

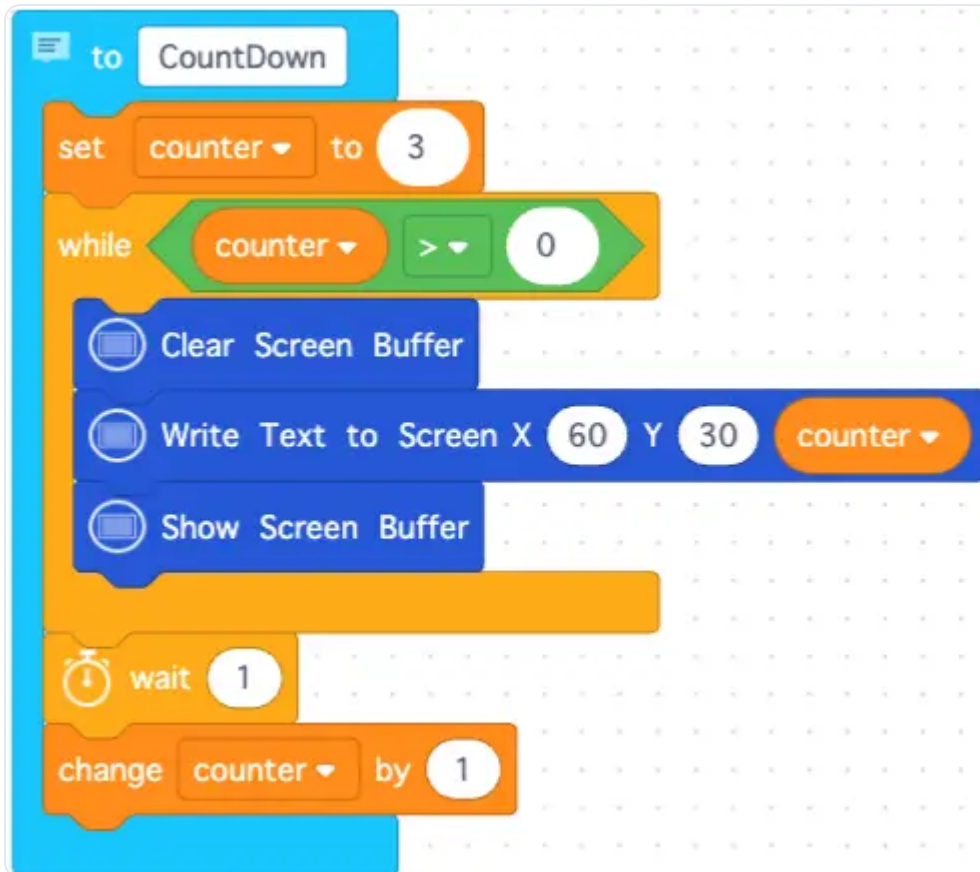
This function shows a countdown on the OLED screen before the game begins.

1. Select **Functions**. Drag a `to do something` block into the code area and drop it.
2. Change **do something** to **CountDown**.
3. Select **Variables**. Select **Create Variable** and type **counter**.

4. Drag a **set counter to** block and snap it inside the **to Countdown** block.
5. Select **Math**. Drag a **0** block and snap it into the blank space on your **set counter to** block.
6. Change the **0** to **3**.
7. Select **Loops**. Drag a **while** block and snap it below the **set counter to 3** block.
8. Select **Logic**. Drag an **=** block and snap it into the blank space on your **while** block.
9. Change the **=** sign to a **>** sign.
10. Select **Variables**. Drag a **counter** block and snap it into the first blank space on your **while** block.
11. Select **Math**. Drag a **0** block and snap it into the other blank space on your **while** block.
12. Select **Display**. Drag a **Clear Screen Buffer** block and snap it inside the **while counter > 0** block.
13. Select **Display**. Drag a **Write Text to Screen** block and snap it under the **Clear Screen Buffer** block.
14. Change the **X** value to **60** and the **Y** value to **30**.
15. Select **Variables**. Drag a **counter** block and snap it into the **PicoBricks** slot on your **Write Text to Screen** block.
16. Select **Display**. Drag a **Show Screen Buffer** block and snap it under the **Write Text to Screen X 60 Y 30 counter** block.
17. Select **Loops**. Drag a **wait** block and snap it below the **Show Screen Buffer** block.
18. Select **Variables**. Drag a **change counter by** block and snap it below the **wait 1** block.
19. Change the **1** to **-1**.



The countdown starts at 3, then changes the counter by -1 each time, so it counts down: 3, 2, 1.



6 Create the Timer function

Next, we will create a function called `TimerFunc`.

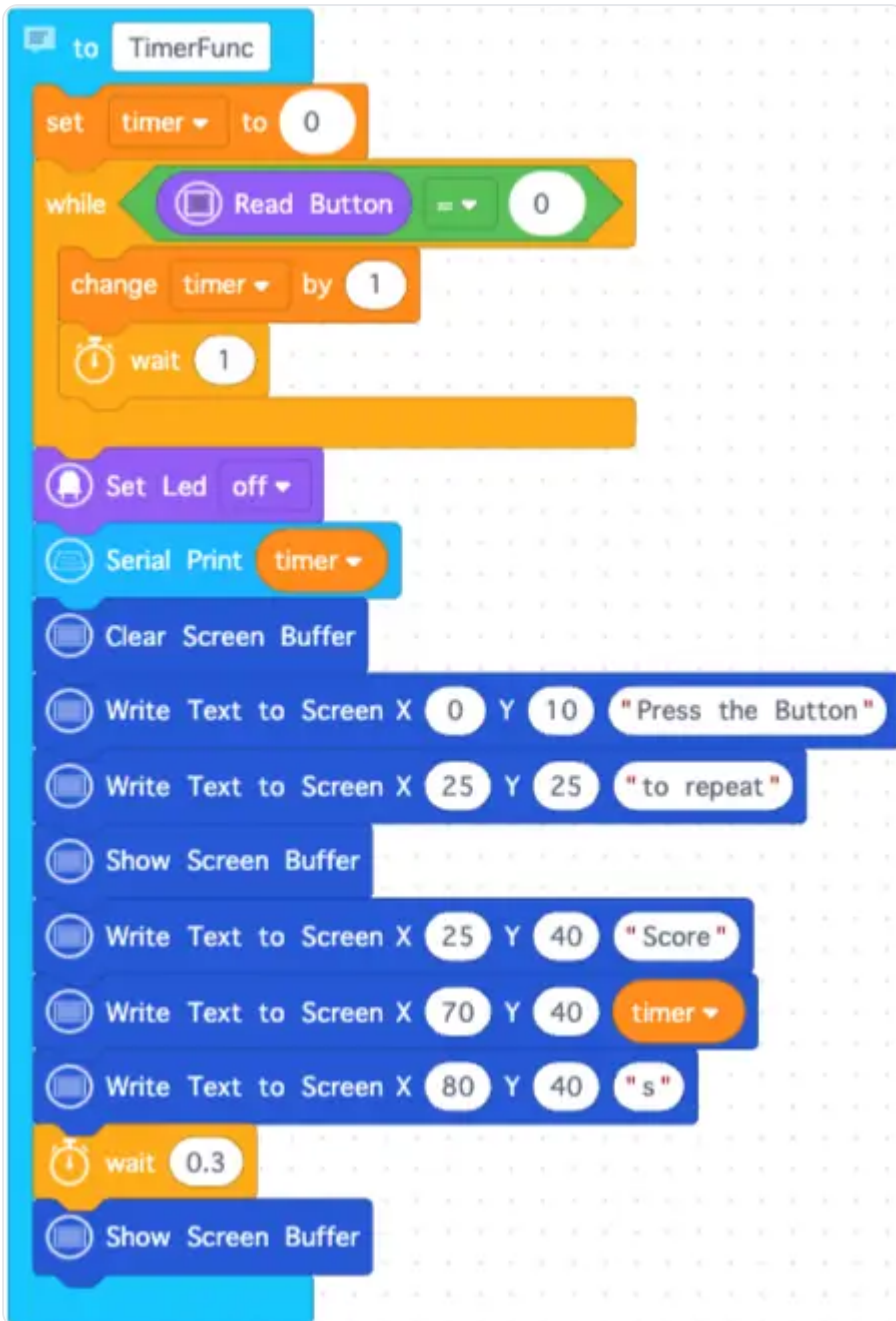
This function measures how long it takes the player to press the button after the LED turns on.

1. Select **Functions**. Drag a `to do something` block into the code area and drop it.
2. Change **do something** to **TimerFunc**.
3. Select **Variables**. Select **Create Variable** and type **timer**.
4. Drag a `set timer to` block and snap it inside the `to TimerFunc` block.
5. Select **Math**. Drag a `0` block and snap it into the blank space on your `set timer to` block.
6. Select **Loops**. Drag a `while` block and snap it below the `set timer to 0` block.
7. Select **Logic**. Drag an `=` block and snap it into the blank space on your `while` block.
8. Select **Bricks**. Drag a `Read Button` block and snap it into the first blank space on your `while` block.
9. Select **Math**. Drag a `0` block and snap it into the other blank space on your `while` block.
10. Select **Variables**. Drag a `change timer by` block and snap it inside the `while Read Button = 0` block.
11. Select **Loops**. Drag a `wait` block and snap it under the `change timer by 1` block.

12. Select **Bricks**. Drag a `Set Led` block and snap it under the `while Read Button = 0` block.
13. Change **on** to **off**.
14. Select **Basic**. Drag a `Serial Print` block and snap it under the `Set Led off` block.
15. Select **Variables**. Drag a `timer` block and snap it into the `Serial Print` block where it says **PicoBricks**.
16. Select **Display**. Drag a `Clear Screen Buffer` block and snap it under the `Serial Print timer` block.
17. Select **Display**. Drag a `Write Text to Screen` block and snap it under the `Clear Screen Buffer` block.
18. Change the **Y** value to **10** and change **PicoBricks** to **Press the Button**.
19. Select **Display**. Drag another `Write Text to Screen` block and snap it underneath.
20. Change the **X** and **Y** values to **25** and change **PicoBricks** to **to repeat**.
21. Select **Display**. Drag a `Show Screen Buffer` block and snap it underneath.
22. Select **Display**. Drag another `Write Text to Screen` block and snap it under the `Show Screen Buffer` block.
23. Change the **X** value to **25**, the **Y** value to **40**, and change **PicoBricks** to **Score**.
24. Select **Display**. Drag another `Write Text to Screen` block and snap it underneath.
25. Change the **X** value to **70** and the **Y** value to **40**.
26. Select **Variables**. Drag a `timer` block and snap it into the **PicoBricks** slot on your `Write Text to Screen` block.
27. Select **Display**. Drag another `Write Text to Screen` block and snap it underneath.
28. Change the **X** value to **80**, the **Y** value to **40**, and change **PicoBricks** to **s** for seconds.
29. Select **Loops**. Drag a `wait` block and snap it underneath.
30. Change **1** to **0.3**.
31. Select **Display**. Drag a `Show Screen Buffer` block and snap it under the `wait 0.3` block.



This function keeps counting until the player presses the button. The quicker they press it, the lower their score will be.

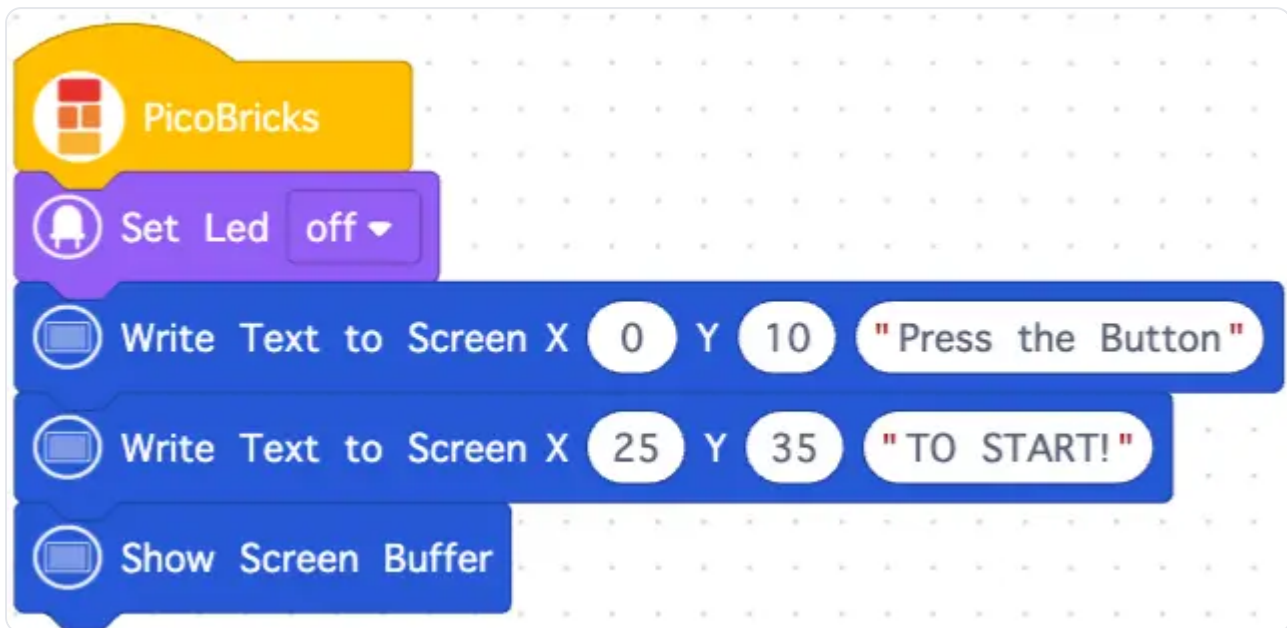


7 Create the game start code

Now we will create the code that starts the game and shows the player what to do.

1. Select **Basic**. Drag a **PicoBricks** block into the code area and drop it.
2. Select **Bricks**. Drag a **Set Led** block and snap it onto the **PicoBricks** block.
3. Change **on** to **off**.
4. Select **Display**. Drag a **Write Text to Screen** block and snap it under the **Set Led off** block.
5. Change the **Y** value to **10** and change **PicoBricks** to **Press the Button**.
6. Select **Display**. Drag another **Write Text to Screen** block and snap it underneath.
7. Change the **X** value to **25**, the **Y** value to **35**, and change **PicoBricks** to **To Start!**

8. Select **Display**. Drag a `Show Screen Buffer` block and snap it underneath.



8 Build the forever loop

Next, we are going to create the forever code that runs our reaction game.

This code will:

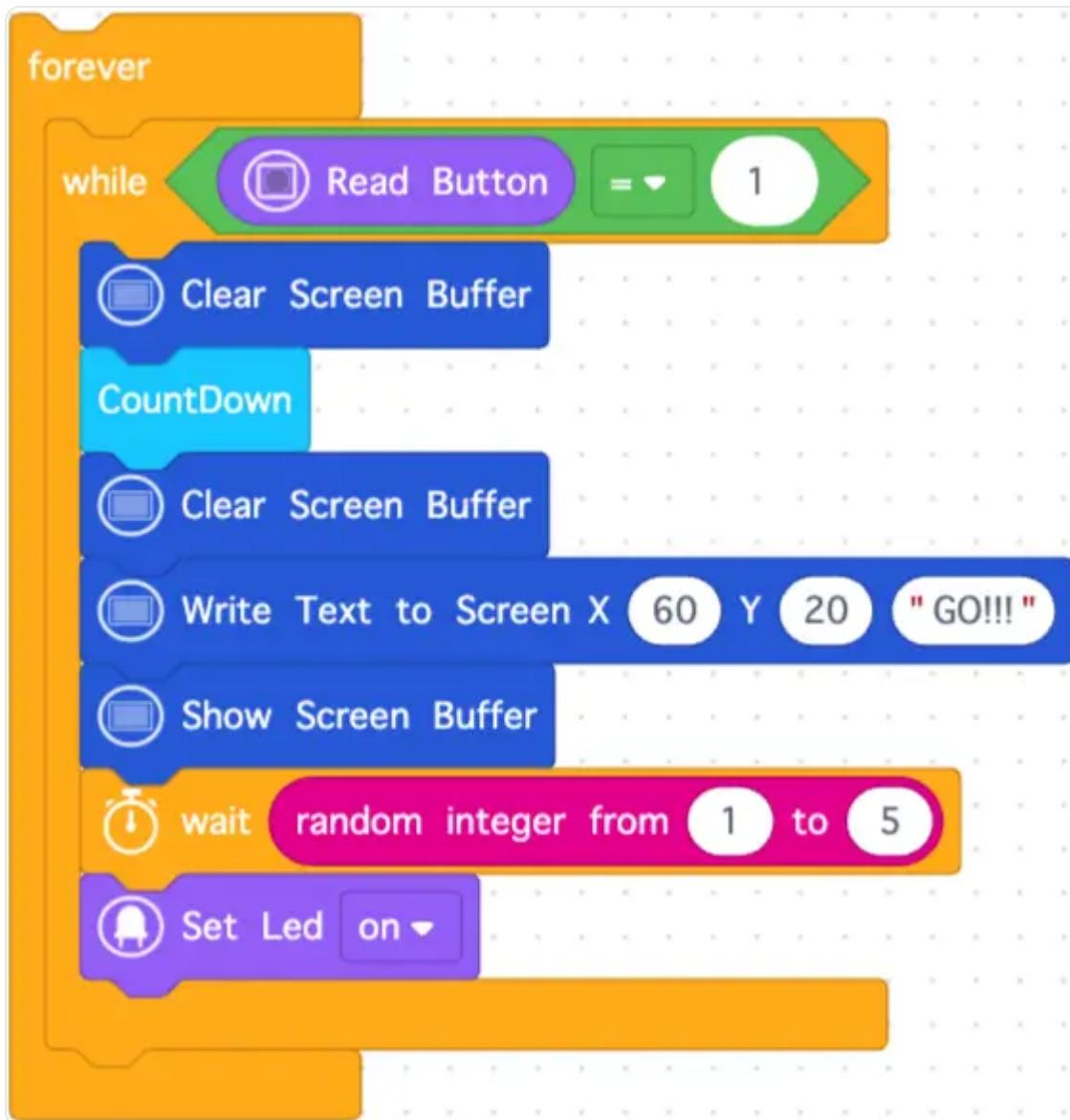
- Clear the OLED screen after the button is pressed
- Start the countdown when the `CountDown` function is called
- Turn the LED on randomly after **Go!!!** has been displayed
- Print the score once we call the `TimerFunc` function

1. Select **Loops**. Drag a `forever` block and snap it under the `Show Screen Buffer` block.
2. Select **Loops**. Drag a `while` block and snap it inside the `forever` block.
3. Select **Logic**. Drag an `=` block and snap it into the blank space on your `while` block.
4. Select **Bricks**. Drag a `Read Button` block and snap it into the first blank space on your `while` block.
5. Select **Math**. Drag a `0` block and snap it into the other blank space on your `while` block.
6. Change the **0** to **1**.
7. Select **Display**. Drag a `Clear Screen Buffer` block and snap it inside the `while Read Button = 1` block.
8. Select **Functions**. Drag a `CountDown` block and snap it below the `Clear Screen Buffer` block.
9. Select **Display**. Drag a `Clear Screen Buffer` block and snap it under the `CountDown` block.

10. Select **Display**. Drag a `Write Text to Screen` block and snap it under the `Clear Screen Buffer` block.
11. Change the **X** value to **60**, the **Y** value to **20**, and change **PicoBricks** to **Go!!!**.
12. Select **Display**. Drag a `Show Screen Buffer` block and snap it below the `Write Text to Screen X 60 Y 20 "Go!!!"` block.
13. Select **Loops**. Drag a `wait` block and snap it below the `Show Screen Buffer` block.
14. Select **Math**. Drag a `random integer from` block and snap it into the **1** slot on your `wait` block.
15. Change the **100** to **5**.
16. Select **Bricks**. Drag a `Set Led` block and snap it under the `wait random integer 1 to 5` block.
17. Select **Functions**. Drag a `TimerFunc` block and snap it under the `Set Led on` block.



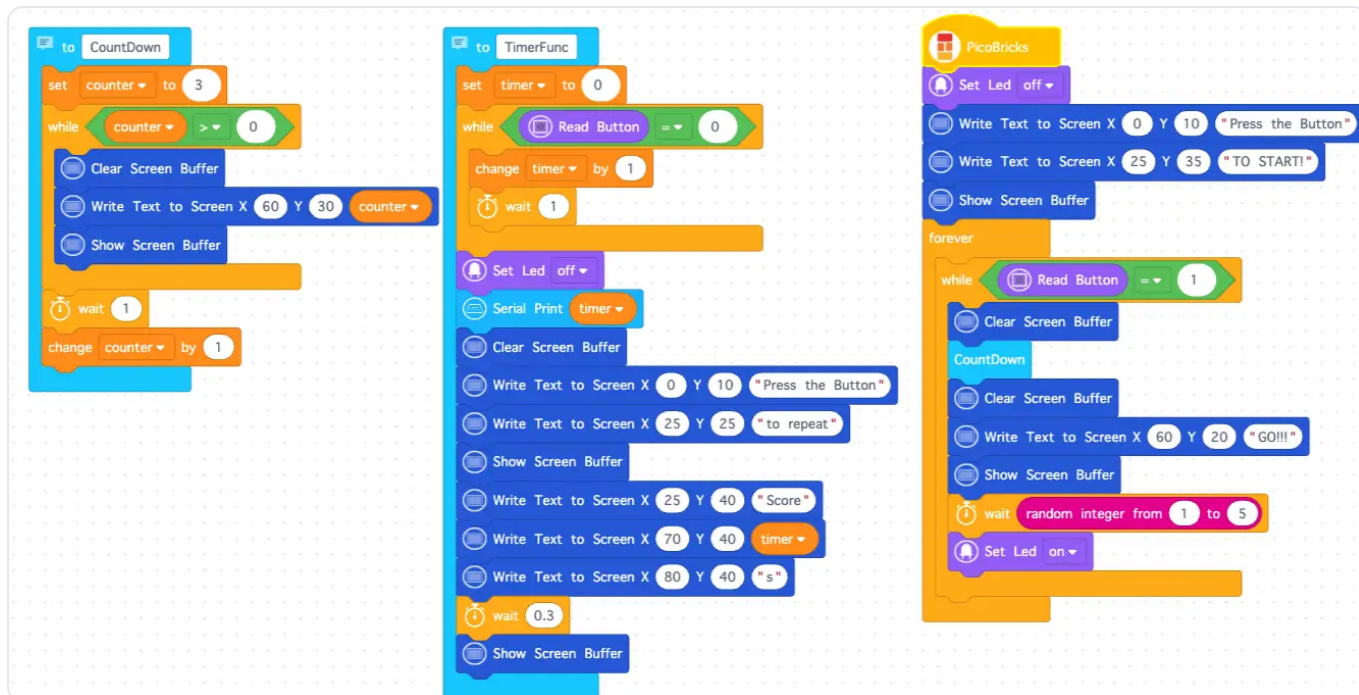
The random wait means the LED will not turn on at the same time every round. That makes the game harder because the player cannot guess when to press the button.



9 Check your completed code

This is now our completed forever loop code and all the code needed to run our reaction time game.

Before you press **Run**, compare your blocks with the completed code below.



10 Run it and test

Now the code is finished, let's test the reaction game.

1. Select the **Run** button at the top of the code area.
2. Press the button to start the game.
3. Watch the countdown on the OLED screen.
4. Wait for the LED to turn on.
5. When the LED turns on, press the button again as fast as you can.
6. Check your reaction time on the OLED screen.
7. Press the button again if you want to repeat the game.



You have built your own reaction-time game with PicoBricks. Now challenge someone nearby and see who can react the fastest!

11 Try it yourself



Try it yourself

Challenge: Can you make the game even cleverer?

Try adding a buzzer that beeps when the LED lights up, or keep score so you can play **best of 3** against a friend.

12 Stuck? Quick fixes

Remember: mistakes are part of learning!

Problem	Try this
The screen stays blank	Check the OLED screen is plugged in properly, and that you added a <code>Show Screen Buffer</code> block after writing your text.
My code won't connect	Make sure the USB cable is pushed in all the way, then click the Connection button to pair your Pico.
The LED never lights up	Check your <code>Set Led</code> block says on (not off) in the forever loop, and that the LED module is connected.
The timer keeps counting forever	Remember to press the button after the LED lights up, as that's what tells the timer to stop.