TaskMicrobit

```
// Game instruction
//
// Generate Random LED light function ONLY ONE at a time
//
// Check condition if pressing ButtonA, ButtonB and ButtonAB
//
// Round is unlimited until the player loses
//
// If correct, then turn off the LED light that appear, and then
move on to the next round (Count the score)
//
// If wrong, then turn off the LED light and then lose the game
(Display "GameOver" and the final score) or
//
//count the score and time spent playing
//
//Extra features
// After 3 seconds, nothing happens then lose the game
//
// If the round is less than 3 seconds, keep the player playing
the game


// 3 seconds in milliseconds
// let roundDuration = 3000

let startTime = 0;

let randomX = 0;
let randomY = 0;

let score = 0;

let isRoundOver = false

// Playing each ROUND
function resetGame() {

    if (!(isRoundOver)) {

        // Generate a random X-coordinate
        randomX = Math.randomRange(0, 4)
```

```
        // Generate a random Y-coordinate
        randomY = Math.randomRange(0, 4)

        // console.log(randomX);
        // console.log(randomY);

        // Random Generate LED light
        RandomGenerateLEDlight(randomX, randomY);
        // console.log(led.point(randomX, randomY));

        if(startTime == 0) {
          startTime = input.runningTime();
        //   console.log("Start:" + startTime);
        }


            input.onButtonPressed(Button.A, function () {
                if (checkIfLEDexistsColumnA(randomX, randomY)) {
                    led.unplot(randomX, randomY);
                    // Update the score
                    score += 1;
                    isRoundOver = false;
                    // console.log(led.point(randomX, randomY));
                    resetGame();
                } else {
                    let elapsedTime = input.runningTime() -
startTime;
                    led.unplot(randomX, randomY);
                    isRoundOver = true;
                    basic.showString("GameOver! Score:" + score);
                    basic.showNumber(score);
                    //one second = 1000 milliseconds
                    elapsedTime = elapsedTime/1000;
                    // console.log("End: " + elapsedTime);
                    basic.showString("Time:" +
elapsedTime.toString());
                    resetGame();
                }
            })
            input.onButtonPressed(Button.B, function () {
                if (checkIfLEDexistsColumnB(randomX, randomY)) {
                    led.unplot(randomX, randomY);
                    // Update the score
                    score += 1;
                    isRoundOver = false;
                    // console.log(led.point(randomX, randomY));
```

```
                    resetGame();
                } else {
                    let elapsedTime = input.runningTime() -
startTime;
                    led.unplot(randomX, randomY);
                    isRoundOver = true;
                    basic.showString("GameOver! Score:" + score);
                    basic.showNumber(score);
                    elapsedTime = elapsedTime / 1000;
                    // console.log("End: " + elapsedTime);
                    basic.showString("Time:" +
elapsedTime.toString());
                    resetGame();
                }
            })
            input.onButtonPressed(Button.AB, function () {
                if (checkIfLEDexistsColumnAB(randomX, randomY)) {
                    led.unplot(randomX, randomY);
                    // Update the score
                    score += 1;
                    isRoundOver = false;
                    // console.log(led.point(randomX, randomY));
                    resetGame();
                } else {
                    let elapsedTime = input.runningTime() -
startTime;
                    led.unplot(randomX, randomY);
                    isRoundOver = true;
                    basic.showString("GameOver! Score:" + score);
                    basic.showNumber(score);
                    elapsedTime = elapsedTime / 1000;
                    // console.log("End: " + elapsedTime);
                    basic.showString("Time:" +
elapsedTime.toString());
                    resetGame();
                }
            })

    } else {
        basic.clearScreen();  // Clear the LED matrix
        score = 0;  // Reset the score
        startTime = 0; // Reset the startTime
        isRoundOver = false; // Reset 'isRoundOver' variable
        resetGame();
    }
}
```

```
//At the start of the game, countdown 3..2..1..
function countdownStartGame() {
    for (let i = 3; i > 0; i--) {
        basic.showNumber(i);
        basic.pause(500);
        basic.clearScreen();
    }
}

//Show LEDs lights Animation
function showInitialAnimation() {
  basic.showLeds(`
    . . . . .
    . . . . .
    . . . . .
    . . . . .
    . . . . .
  `)

  basic.showLeds(`
    . . # . .
    . . # . .
    # # # # #
    . . # . .
    . . # . .
  `)

  basic.showLeds(`
    . . . . .
    . . . . .
    . . . . .
    . . . . .
    . . . . .
  `)
}

//Light up LEDs function
function RandomGenerateLEDlight(positionX: number, positionY:
number) {
    // led.point(positionX, positionY);
    led.plot(positionX, positionY);
}

//Check the condition if the light on a specific coordinate is
"ON" or "OFF" (all three functions)
```

```
function checkIfLEDexistsColumnA(positionX: number, positionY:
number) {
        //led.point(column, row)
        //led.point(0, 0) || led.point(1, 0) || led.point(0, 1)
|| led.point(1, 1) || led.point(0, 2) || led.point(1, 2) ||
led.point(0, 3) || led.point(1, 3) || led.point(0, 4) ||
led.point(1, 4)
        if (led.point(0, 0) || led.point(1, 0) || led.point(0, 1)
|| led.point(1, 1) || led.point(0, 3) || led.point(1, 3) ||
led.point(0, 4) || led.point(1, 4)) {
            return true;
        }
    return false
}

function checkIfLEDexistsColumnB(positionX: number, positionY:
number) {
        //led.point(3, 0) || led.point(4, 0) || led.point(3, 1)
|| led.point(4, 1) || led.point(3, 2) || led.point(4, 2) ||
led.point(3, 3) || led.point(4, 3) || led.point(3, 4) ||
led.point(4, 4)
        if (led.point(3, 0) || led.point(4, 0) || led.point(3, 1)
|| led.point(4, 1) || led.point(3, 3) || led.point(4, 3) ||
led.point(3, 4) || led.point(4, 4)) {
            return true;
        }
    return false
}

//(0, 2) (1, 2) (3, 2) (4, 2)
function checkIfLEDexistsColumnAB(positionX: number, positionY:
number) {
        //led.point(2, 0) || led.point(2, 1) || led.point(2, 2)
|| led.point(2, 3) || led.point(2, 4)
        if (led.point(2, 0) || led.point(2, 1) || led.point(2, 2)
|| led.point(2, 3) || led.point(2, 4) ||
            led.point(0, 2) || led.point(1, 2) || led.point(3, 2)
|| led.point(4, 2)) {
            return true;
        }
    return false
}

// ...3..2..1
countdownStartGame();
```

```
showInitialAnimation();
resetGame();
```

Extensions

- radio, *
- microphone, *