

Lesson 12: More User Input - Keyboard Movement

Lesson aim:

To control things with the keyboard, like we will in our project game.

Why:

We need to know how to process keyboard user input. In term 2 we go further, bringing in typing (just like most apps/programs - WhatsApp, Web browsers, Microsoft Word, the Processing app, etc.)

What user input have we used already?

```
ellipse(mouseX,mouseY, 100,100); and if(mousePressed == true){
```

If you can't remember what the snippets above do, now is the time to go and find out! Check the reference guide, ask another student, check your notes - or if all else fails, ask a teacher!

Now we want to add another input option to our toolkit - the keyboard.

Getting input from the keyboard

Like `mousePressed` there is also `keyPressed`. This is *true* when any key is pressed.

Test your knowledge!:

Make a program with a shape, (any shape you want) that only moves (remember how to make a shape move?) when you press any keyboard key. Show a teacher when you've completed this.

If you are finished early:

Extend the program so that the shape moves a different direction when a mouse button is pressed. So the shape will go one way if a key is pressed and another if the mouse is pressed.

Individual keys:

We can check for individual key presses (in this example key for 'd') using:

```
if(key == 'd'){ //← check if the letter d is pressed
```

Note: the key letter must be in *single* quotes (these: ` `)

Example code we will be working with:

```
int squareHorizontalPoz = 250; //number variable, with value 250
void setup() {
  size(600, 300);
}

void draw() {
  background(0);
  rect(squareHorizontalPoz, 100, 100, 50);
  if(keyPressed == true){ //← check if any key is pressed
    if(key == 'd'){ //← check if the letter d is pressed
      squareHorizontalPoz = squareHorizontalPoz+2;
    }
  }
}
```

Task

1. Copy the above and see what it does.
2. Extend the program so the following happens:
 - a. the 'a' button moves the shape **left**
 - b. the 'w' button moves the shape **up**
 - c. the 's' button moved the shape **down**

Extra tasks:

1. Add extra functionality so that there are other (new) letters that change:
 - a. the colour of the shape to:
 - i. red
 - ii. green
 - iii. yellow
 - b. Change the background colour of the screen to:
 - i. black
 - ii. white
 - iii. blue
 - c. Move the shape:
 - i. automatically back to the centre of the screen (press the key once and the shapes moves instantly to the middle of the screen)
2. Make a new program and draw a triangle. Add code so that you can control the position of each corner individually (different keys move the top, left and right corners of the triangle, so you can move each corner on its own.)