MODULE 4: JUMPING AND CONSTANTS

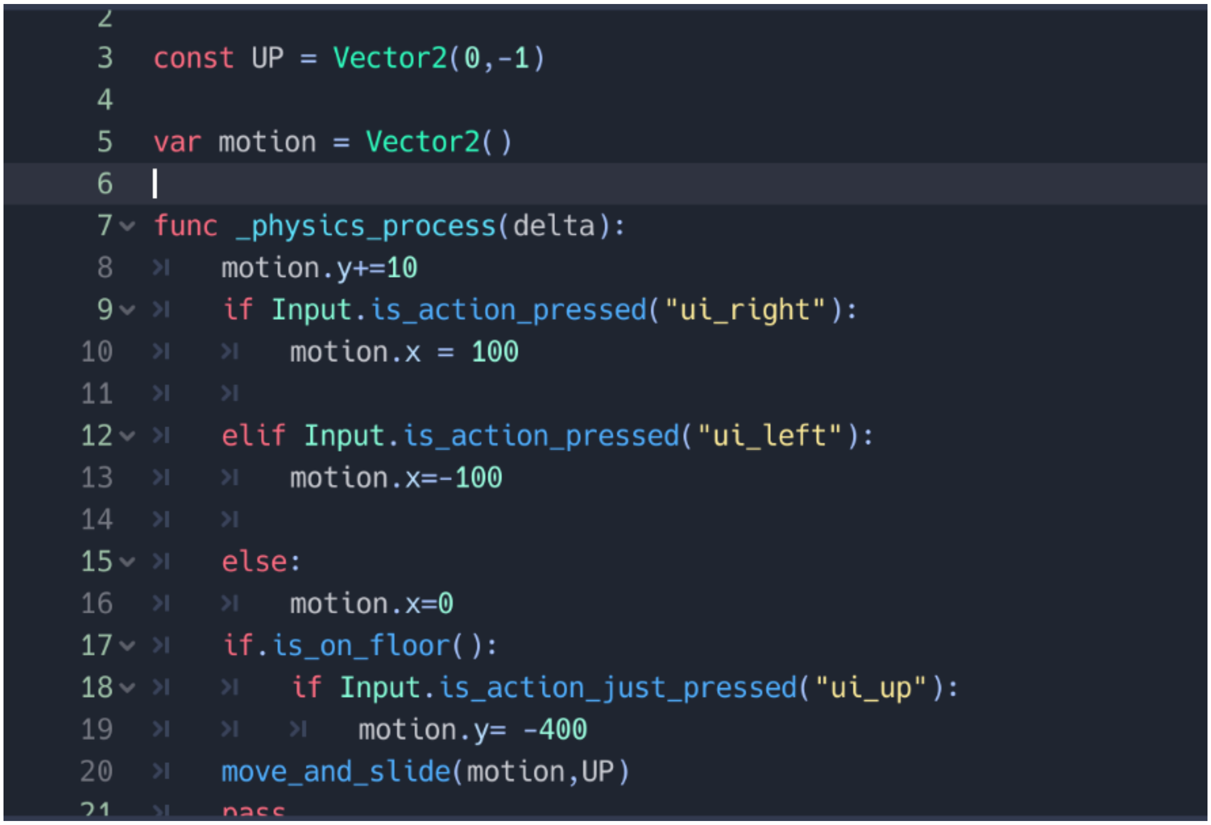
We created a vertical axis in your world, so let's teach the Player how to "jump". Jumping means that the Player can move up and down.

JUMPING

First we'll need to alter the code to tell the Player to jump. We'll also need to define which way is up.

1. Click on the Script icon next to "Player".
2. Add the following after line 1 "extends KinematicBody2D"
   const UP = Vector2(0,-1)
   a. This will define which way is up.
3. After the "else" motion, we also need to add our action of jumping:
   If.is_on_floor():
   If Input.is_action_just_pressed("ui_up"):
   Motion.y = -400
4. Add our Constant to all movement, so edit "move_and_slide(motion)" to:
   move_and_slide(motion,UP)

After adding these new lines, your code should now look like this. Save, and test it out!



```
const UP = Vector2(0,-1)

var motion = Vector2()

func _physics_process(delta):
    motion.y+=10
    if Input.is_action_pressed("ui_right"):
        motion.x = 100

    elif Input.is_action_pressed("ui_left"):
        motion.x=-100

    else:
        motion.x=0
    if.is_on_floor():
        if Input.is_action_just_pressed("ui_up"):
            motion.y= -400
    move_and_slide(motion,UP)
    pass
```

PRO TIP:

- What is a Constant? Constants are consistent, which means that instead of scattering variables all over the code, we can use the same variables throughout.
- Constants are expressed using uppercase letters
- Example: Instead of "-100" for motion, you can change it to "-SPEED". This way you only need to change it once, rather than changing it each time you put it in the code!

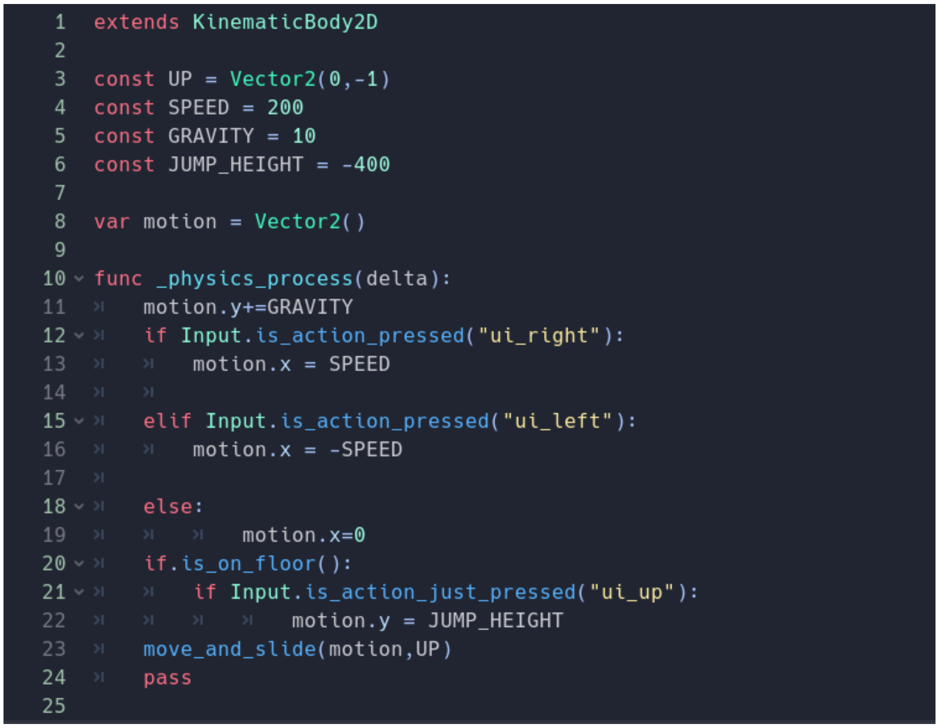
CREATING CONSTANTS

Let's add a constant to our code! Instead of "-100" for motion, let's change it to "-SPEED".

1. Click on the Script icon next to "Player".
2. After const UP, let's add some more constants.

   const SPEED = 100
   const GRAVITY = 10
   const JUMP_HEIGHT = -400
3. Change all your variables to these constants.
   a. Example: motion.x = 100 → motion.x = SPEED

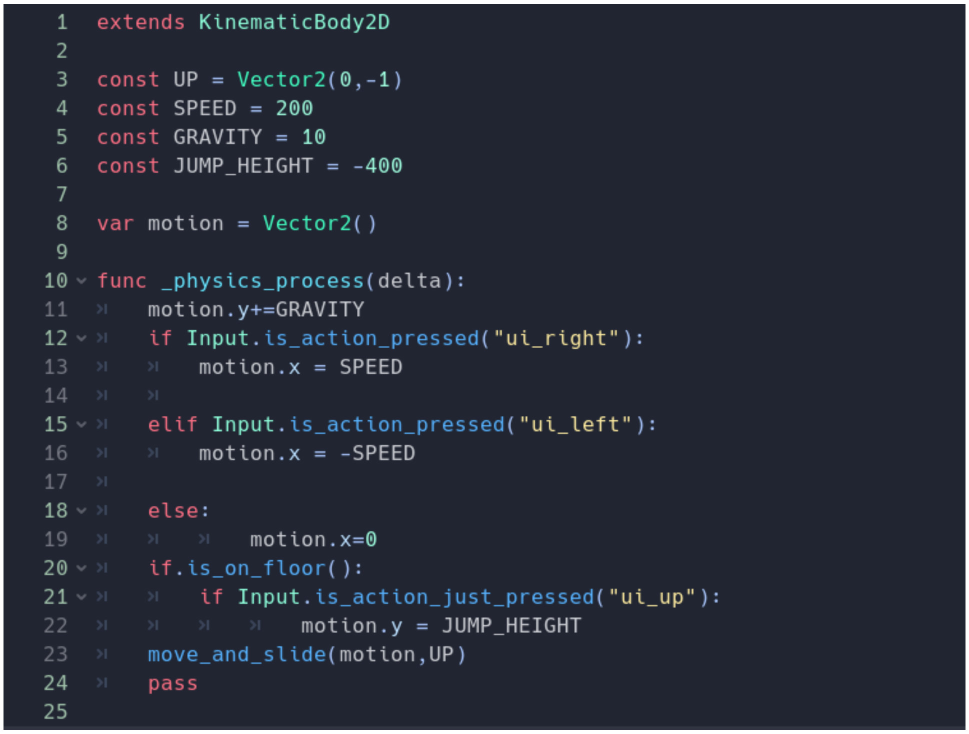
After adding these new lines, your code should now look like this. Save, and test it out! It should work the same, but constants make it easier to do the coding (e.g. you don't have to type 100 every time, you can type SPEED instead).



```
extends KinematicBody2D

const UP = Vector2(0,-1)
const SPEED = 200
const GRAVITY = 10
const JUMP_HEIGHT = -400

var motion = Vector2()

func _physics_process(delta):
    motion.y+=GRAVITY
    if Input.is_action_pressed("ui_right"):
        motion.x = SPEED

    elif Input.is_action_pressed("ui_left"):
        motion.x = -SPEED

    else:
            motion.x=0
    if.is_on_floor():
        if Input.is_action_just_pressed("ui_up"):
                motion.y = JUMP_HEIGHT
    move_and_slide(motion,UP)
    pass

```

FINAL CODE

This is what the final code should look like:



```
extends KinematicBody2D

const UP = Vector2(0,-1)
const SPEED = 200
const GRAVITY = 10
const JUMP_HEIGHT = -400

var motion = Vector2()

func _physics_process(delta):
    motion.y+=GRAVITY
    if Input.is_action_pressed("ui_right"):
        motion.x = SPEED
        
    elif Input.is_action_pressed("ui_left"):
        motion.x = -SPEED
    
    else:
            motion.x=0
    if.is_on_floor():
        if Input.is_action_just_pressed("ui_up"):
            motion.y = JUMP_HEIGHT
    move_and_slide(motion,UP)
    pass

```

WHERE CAN I GO FROM HERE?

In under 25 lines of code, you just created a video game! Obviously this one is a little basic, but what other things do you think you could program the Player to do? There are also plenty of other elements that you can add to your 2D game, including animation and lighting. Once you've mastered that, Godot also has the tools to create 3D games with realistic looking animation!

Godot uses GDScript as its programming language. But once you become familiar with its API (Application Programming Interface), you'll learn a lot of good programming principles that will be helpful in your game and web development journey! Step up your "game" by learning C#, C++, or any of the other popular programming language!

Take me to more Game Development with Godot modules!
Take me to the Godot Video Tutorial!
Comments, questions, or feedback? Email us at academy@geeksquad.ca.
Take me back to Geek Squad Academy Online Learning!