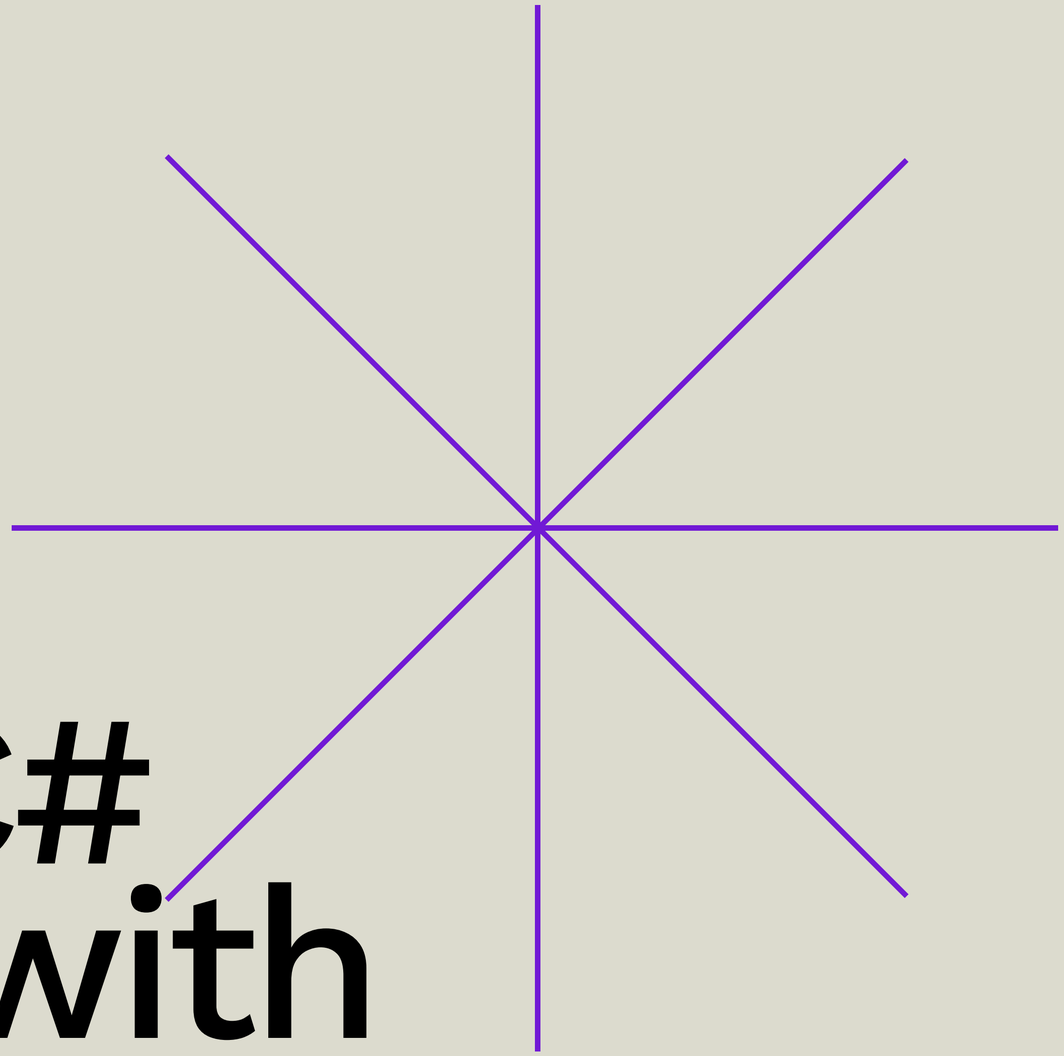
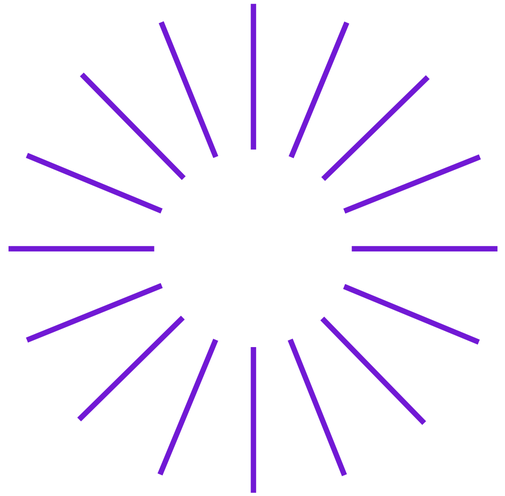


05. Basics of C# Programming with Unity

DANIELA MONTEIRO

INDEX

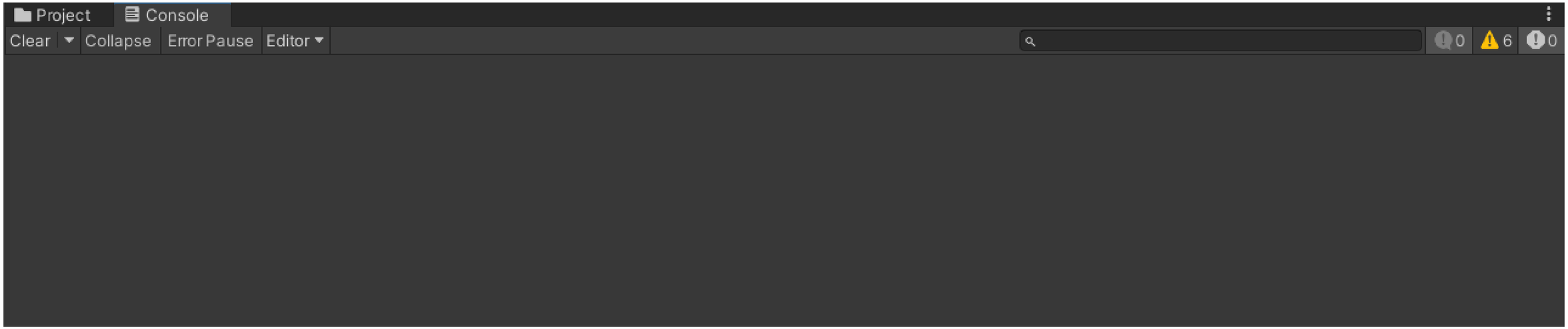


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 referências
public class MyFirstScript : MonoBehaviour
{
    // Start is called before the first frame update
    0 referências
    void Start()
    {

    }

    // Update is called once per frame
    0 referências
    void Update()
    {

    }
}
```

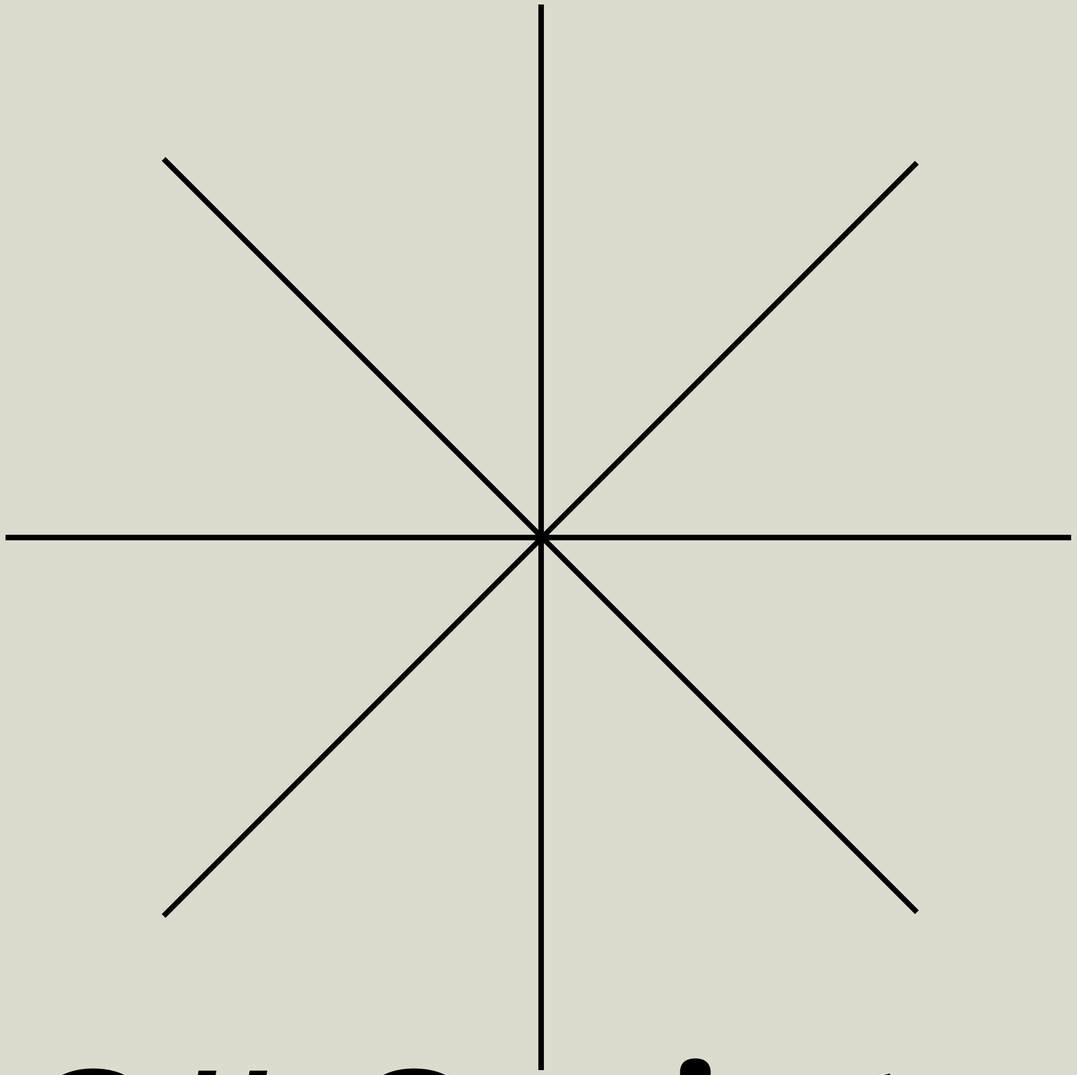
01. WHAT IS A SCRIPT?

A script is a set of instructions that make a certain program execute a determined function.

02. UNITY'S CONSOLE



The most important tab in the Unity Editor for programming. It is through the console that we are able to detect errors in our code and even print messages.



03
Structure of a C# Script

DANIELA MONTEIRO

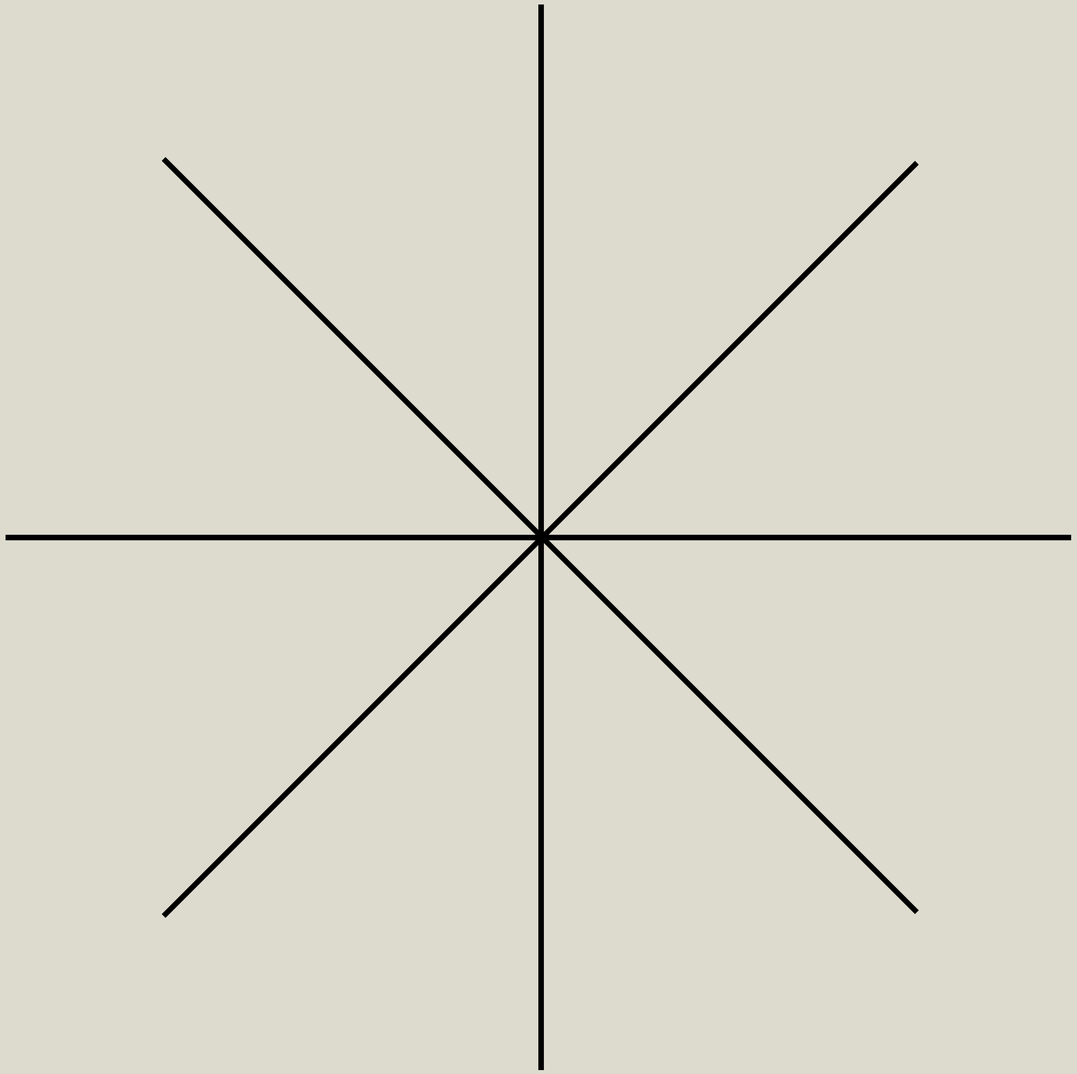
Imported libraries:

The main C# class (MyFirstScript):

The Start method inside our class: (Will only be called once, at the beginning of our game)

The Update method: (It is constantly running, as it is called once every frame)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 referências
public class MyFirstScript : MonoBehaviour
{
    // Start is called before the first frame update
    0 referências
    void Start()
    {

    }

    // Update is called once per frame
    0 referências
    void Update()
    {

    }
}
```



04
Comments

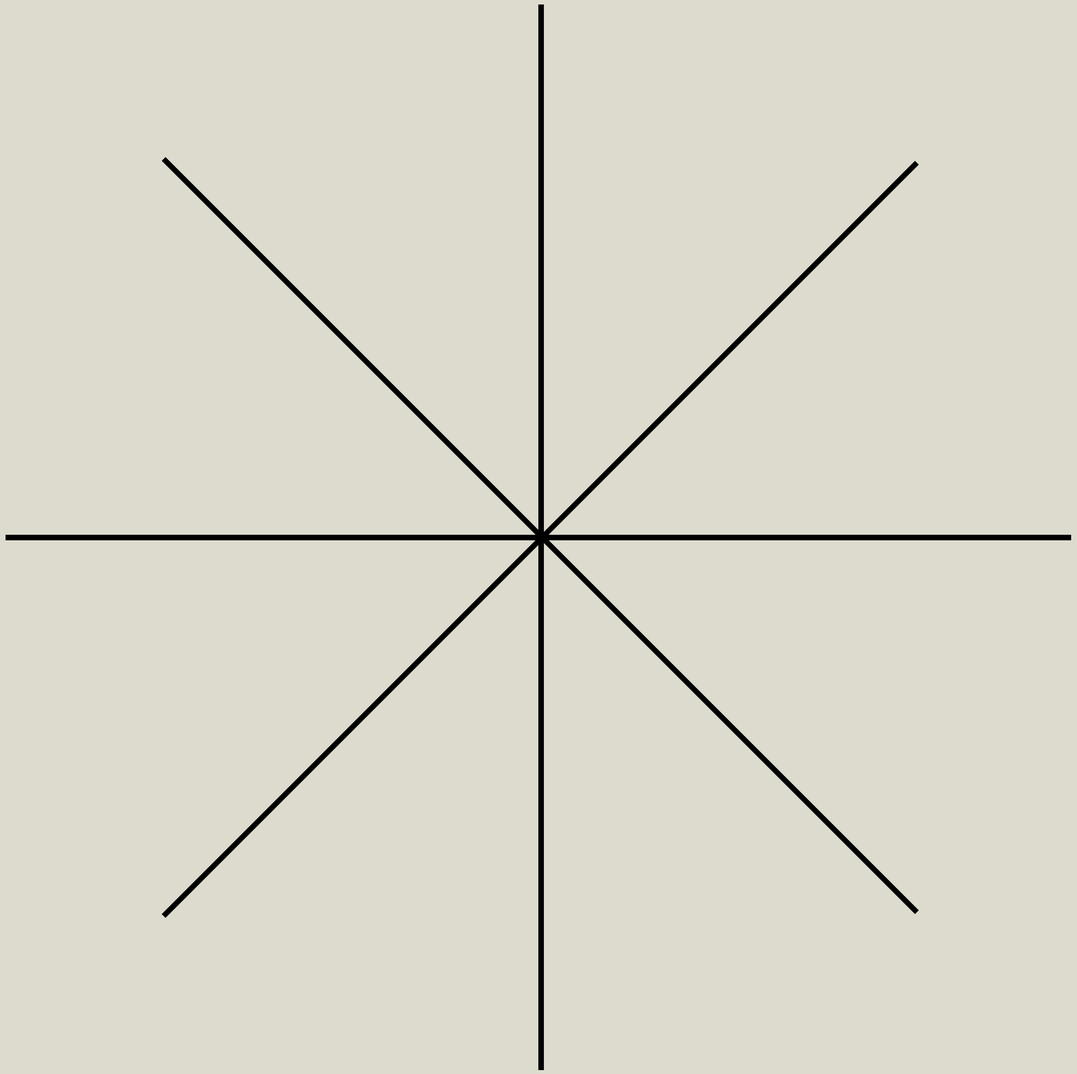
DANIELA MONTEIRO

CTRL + K + C

```
//indicaçao do seguinte codigo:
/*
já não preciso deste for
for (int i = 0; i < 10; i++)
{
    myBool = false;
    myWholeNumber++;
    print("Hello World!");
}
*/
```

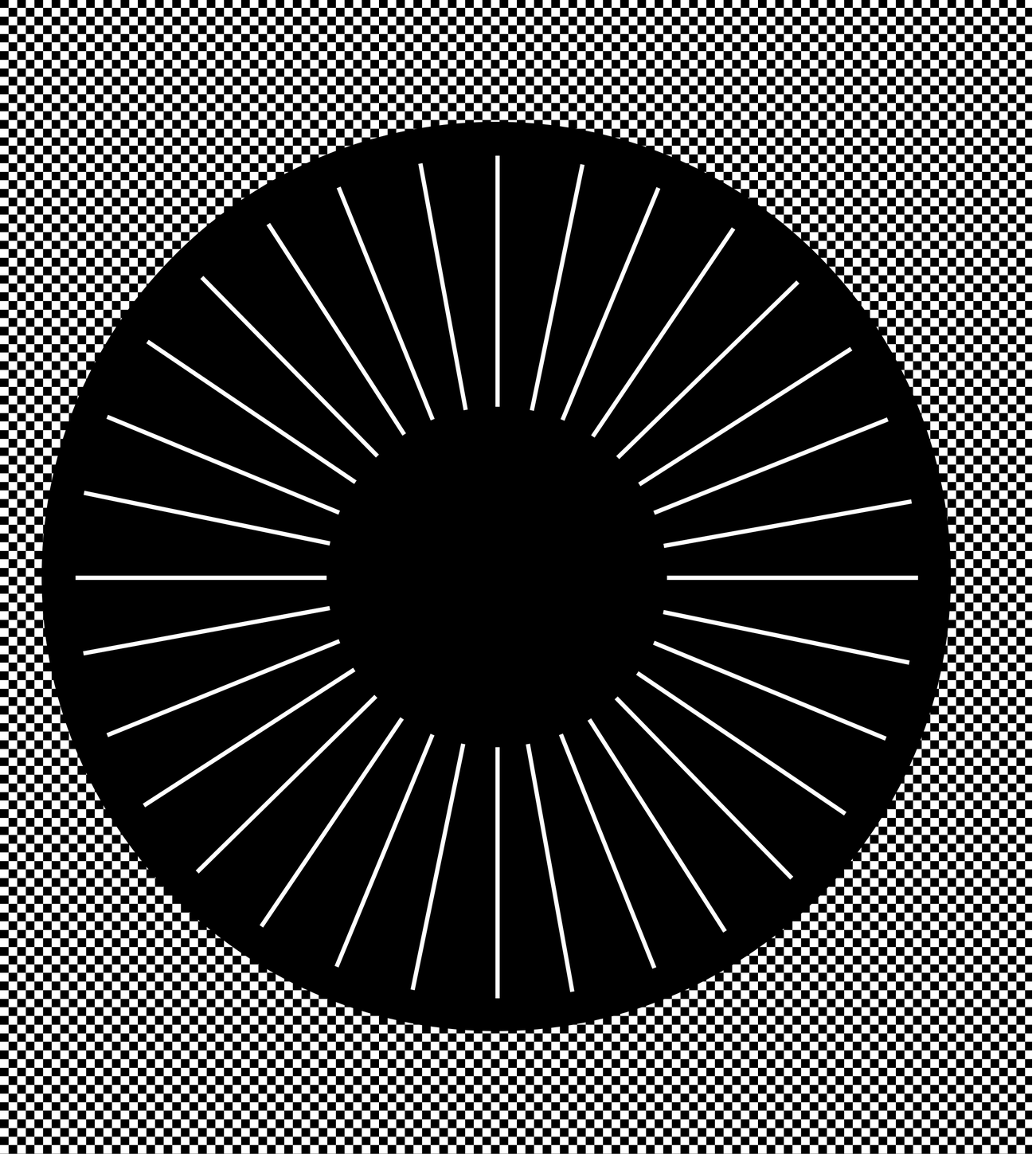
For example to indicate a functions' purpose. Anything that is commented will not be excecuted.

ctrl + k + c to comment
ctrl + k + u to uncomment



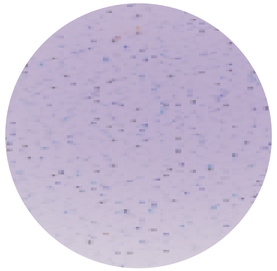
05
Variables

DANIELA MONTEIRO



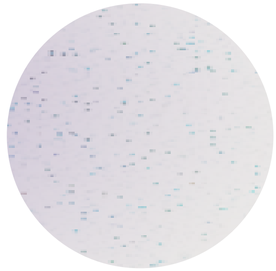
A VARIABLE IS A WAY TO STORE INFORMATION

TYPES OF VARIABLES



int

For whole numbers



double

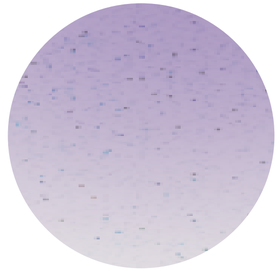
For decimals

char

For single characteres, such as ‘a’ or ‘B’.

string

For strings of text, sunch as “Hello!”



bool

For boolean values (true or false)

To exist in our script, a variable must be initialized before any method.

To be instantiated, you must indicate it’s type and it’s name(it is good practice to name variables following the “camelCase” denomination).

```
public class MyFirstScript : MonoBehaviour
{
    int myWholeNumber;
    double myDecimal;
    char myChar;
    string myString;
    bool myBool;


    // Start is called before the first fra
    0 referências
    void Start()
```

After being instantiated, a value can now attributed to it.

```
void Start()
{
    myWholeNumber = 0;
    myDecimal = 0.1;
    myChar = 'a';
    myString = "Hello World!";
    myBool = true;
}
```

In a C# class, a variable can be public or private.

If it is public it can be acessed from any other class/script in our project.

If it is private, it is only acessible from the script it was initialized in,

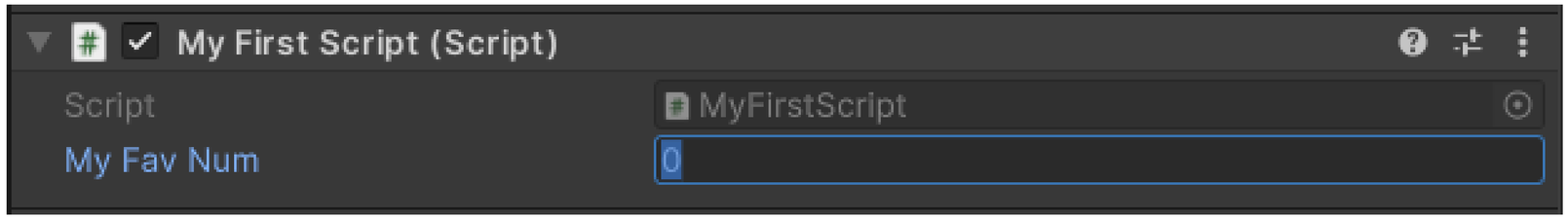
By default, a variable is private.

```
public string myName = "John Doe, anyone can know!";
private string mySalary = "12€ :(, no one can know!";
```

[SERIALIZEFIELD]

```
[SerializeField] public int myFavNum;
```

By making a variable serializable, it allows us to change it’s value from the Unity Editor.



UNITY'S TYPES OF VARIABLES

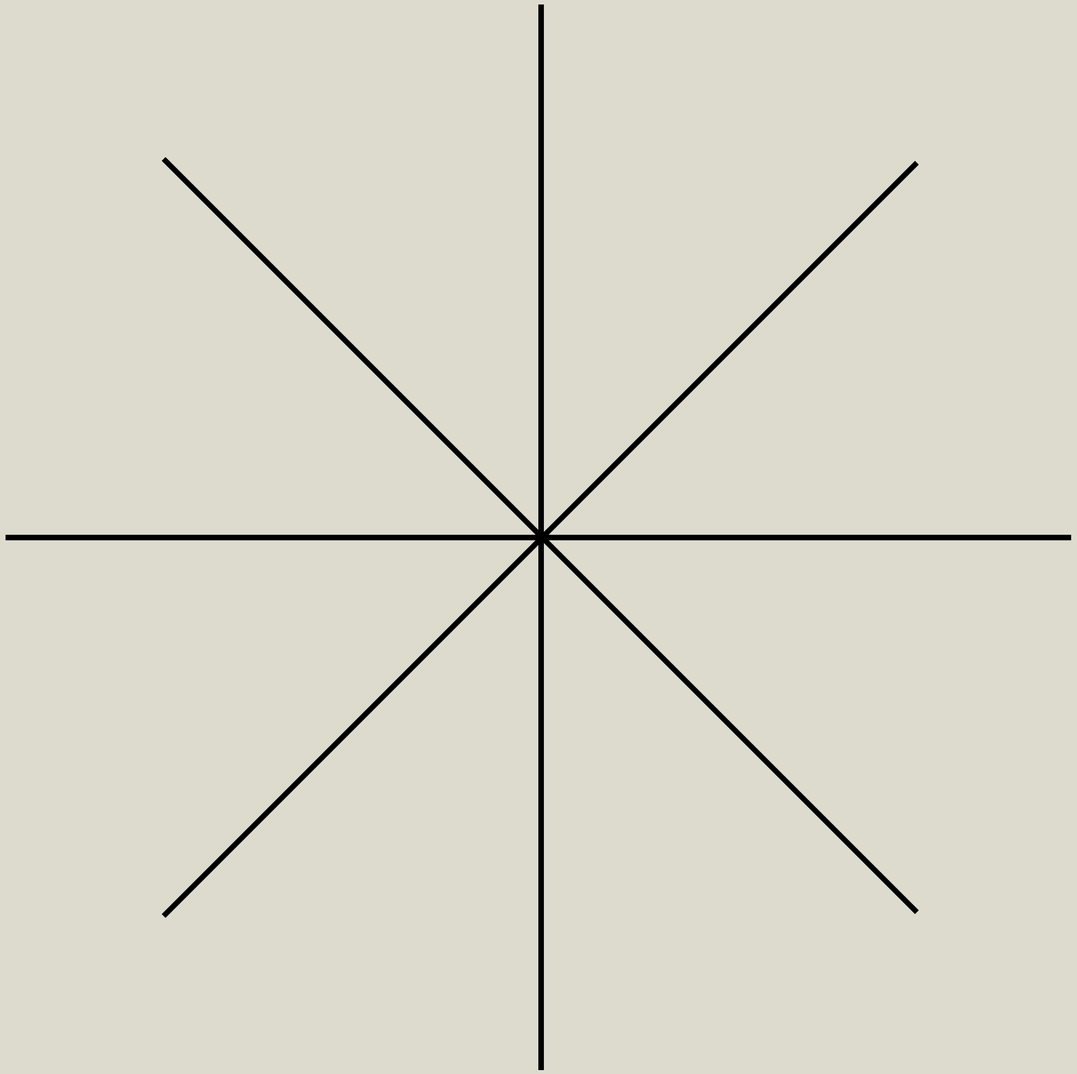
```
GameObject playerObject;
Transform playerTransform;
Animator playerAnimator;
Rigidbody2D playerRigidBody;
```

There are certain types that are specific to Unity.

UNITY'S TYPES OF VARIABLES

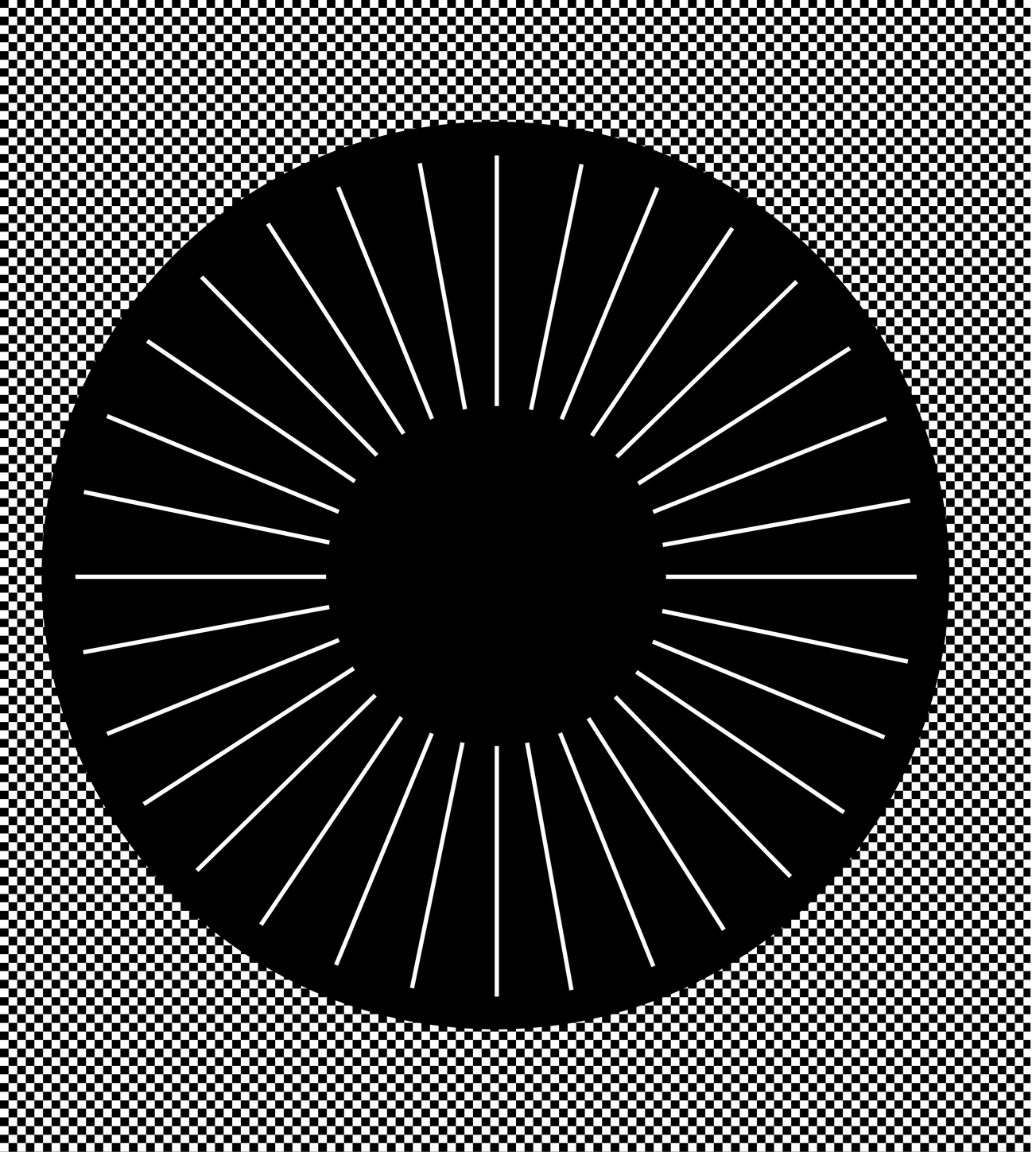
These types of variables are tipically related to the components of the gameObjects in our scene.
To assign them a value it is necessary to get the corresponding component.

```
playerObject = this.gameObject;
playerTransform = playerObject.transform;
playerAnimator = playerTransform.GetComponent<Animator>();
playerRigidBody = playerTransform.GetComponent<Rigidbody2D>();
```



06
Methods

DANIELA MONTEIRO

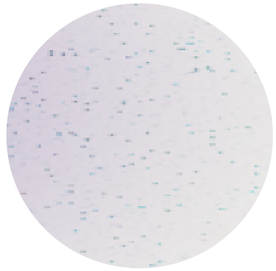


A METHOD IS A SET OF INSTRUCTIONS THAT IS ONLY EXCUTED ONCE THE METHOD IS CALLED.

06. METHODS

Just like variables, a method can be public or private and also has types, which relates to what it returns.

TIPOS DE MÉTODOS



void

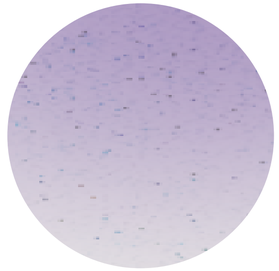
Doesn’t return anything

int

Only returns int values

string

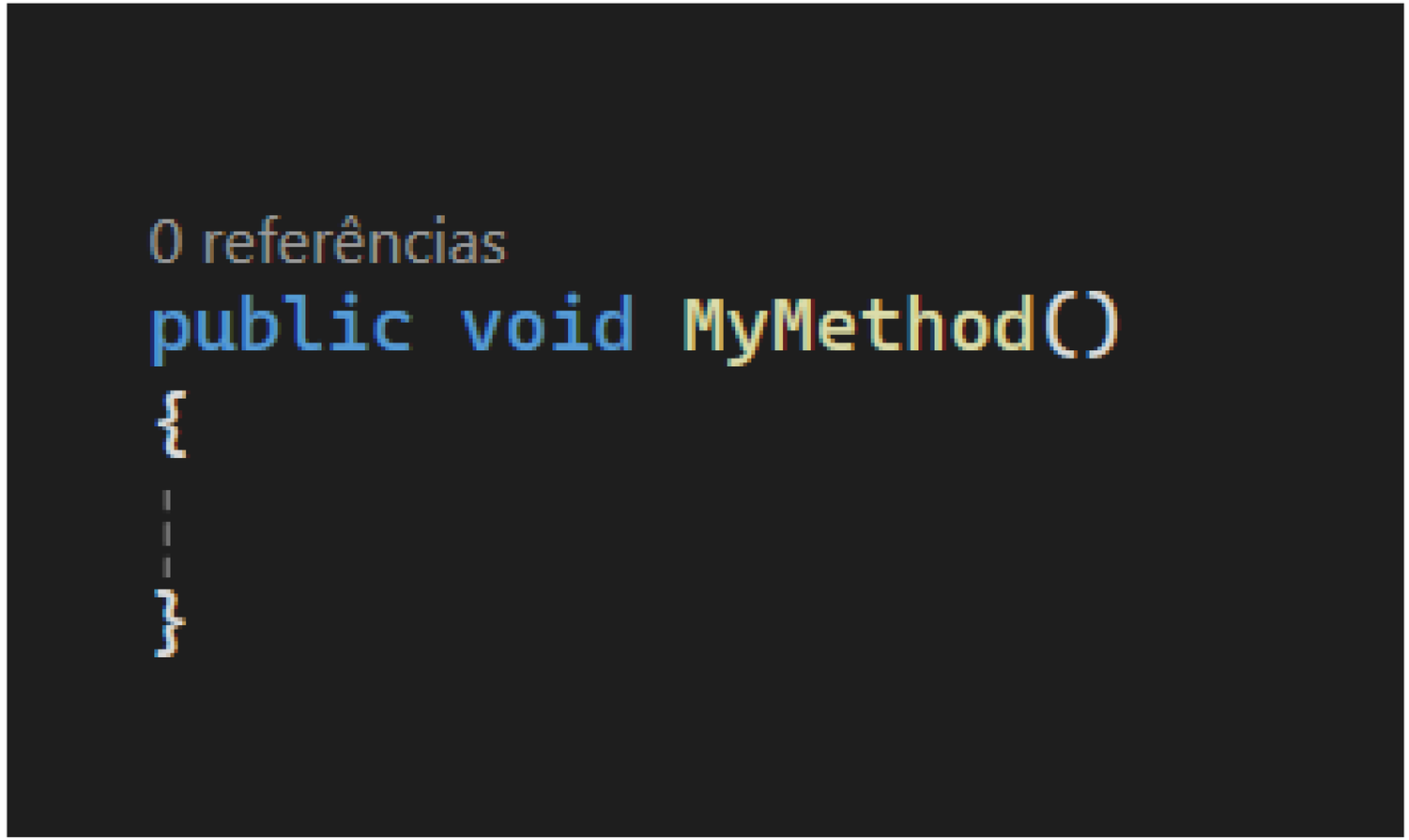
Only returns string values



bool

Only returns boolean values

HOW TO CREATE A METHOD



The name of a method must be capitalized.

HOW TO CALL A METHOD

```
Mensagem do Unity | 0 referênci
void Start()
{
    MyMethod();
}
```

A method will only run once it is called.

METHOD OF THE TYPE INT

```
1 referência
public int MyMethod()
{
    return 0;
}
```

Any method, not of the type void must return a value.

METHOD OF THE TYPE INT

To be called it will attribute it’s returned value to the varible myWholeNumber.

```
Mensagem do Unity | 0 referências
void Start()
{
    myWholeNumber = MyMethod();
}
```

INPUT DATA IN METHODS

```
1 referência
public int MyMethod(int anyNumberIwant)
{
    return anyNumberIwant;
}
```

It is possible to work with a certain varible inside a method

INPUT DATA IN METHODS

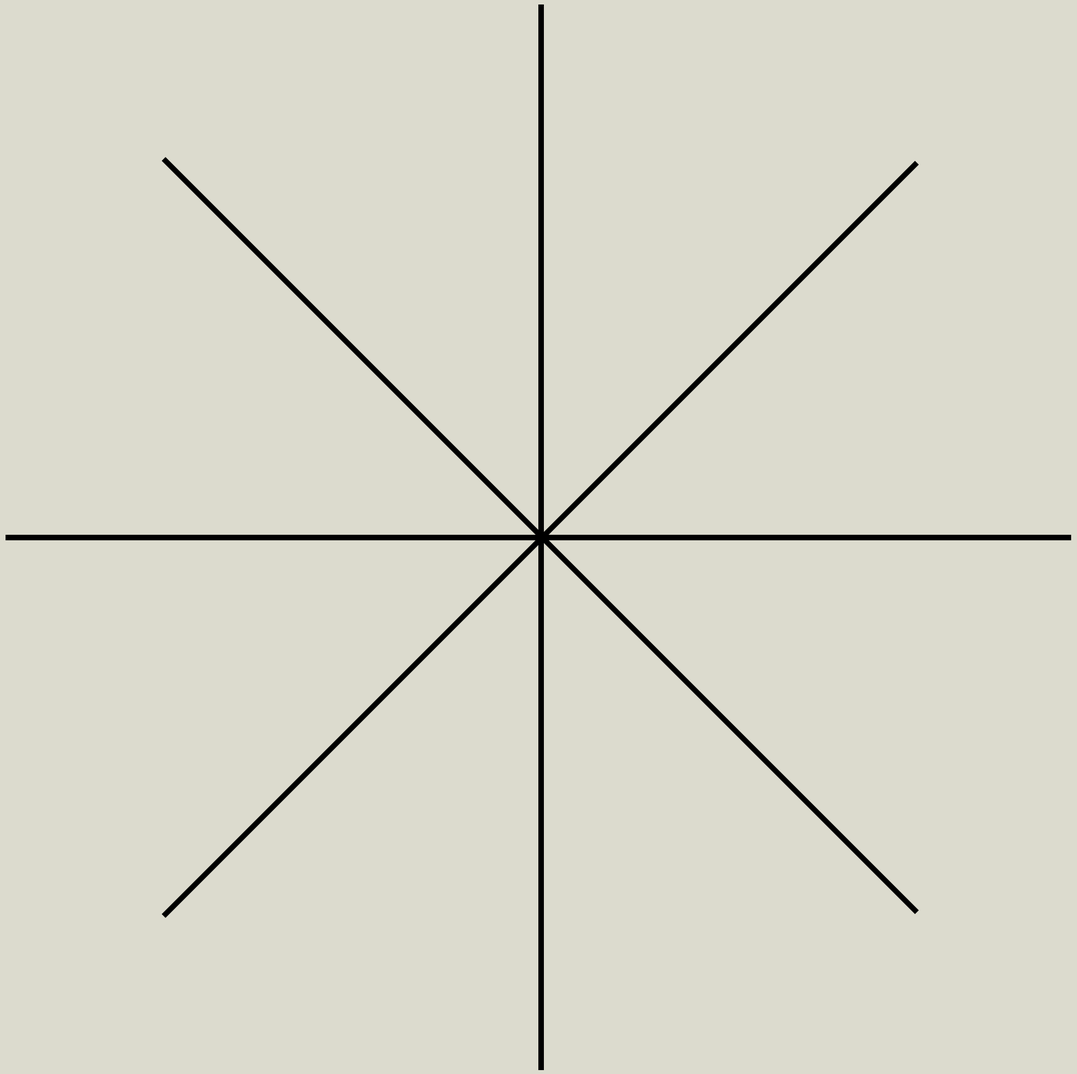
When it is called, a method must be given the necessary parameters (in this case it requires a whole number).

```
Mensagem do Unity | 0 referências
void Start()
{
    myWholeNumber = MyMethod(5);
}
```

MÉTODO PRINT

```
print("Hello World!");
print(myFavNum);
print("This is my fav number: " + myFavNum);
```

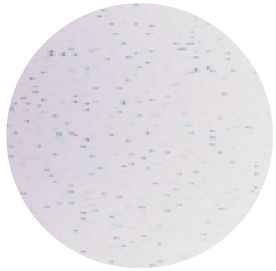
A method that allows any type of information to be printed onto the console.



07
Operators

DANIELA MONTEIRO

OPERADORES MATEMÁTICOS



==

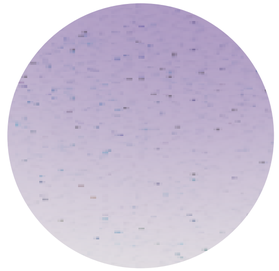
equals

!=

different

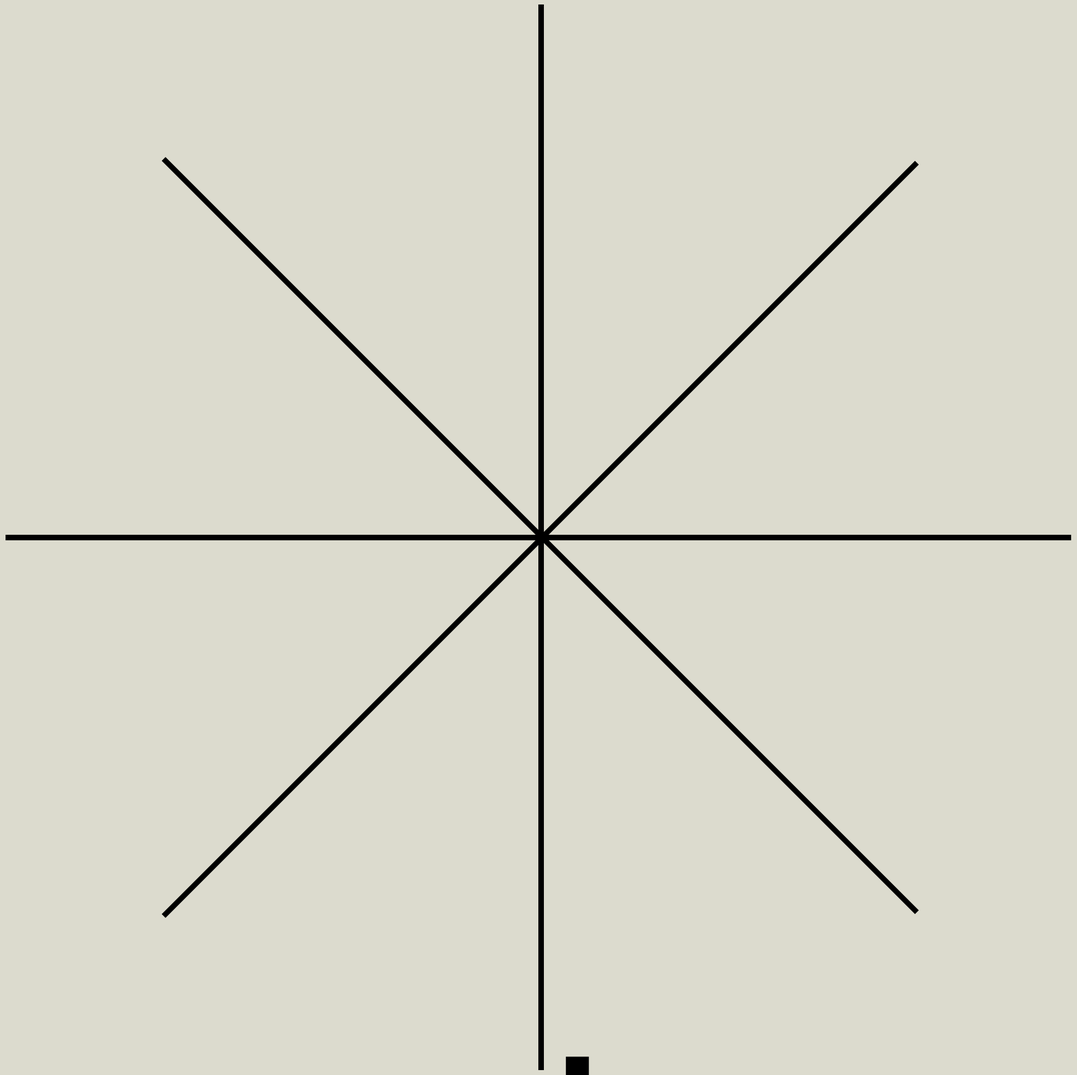
&&

AND



||

OR



08
If and For Commands

DANIELA MONTEIRO

COMANDO IF

O ‘if’ é uma condição lógica cujo seu código só é executado caso satisfaça as condições necessárias.

Neste caso, se a variavel ‘myFavNum’ for maior que 5.

```
if(myFavNum > 5)
{
    print("maoir que 5!");
}
```

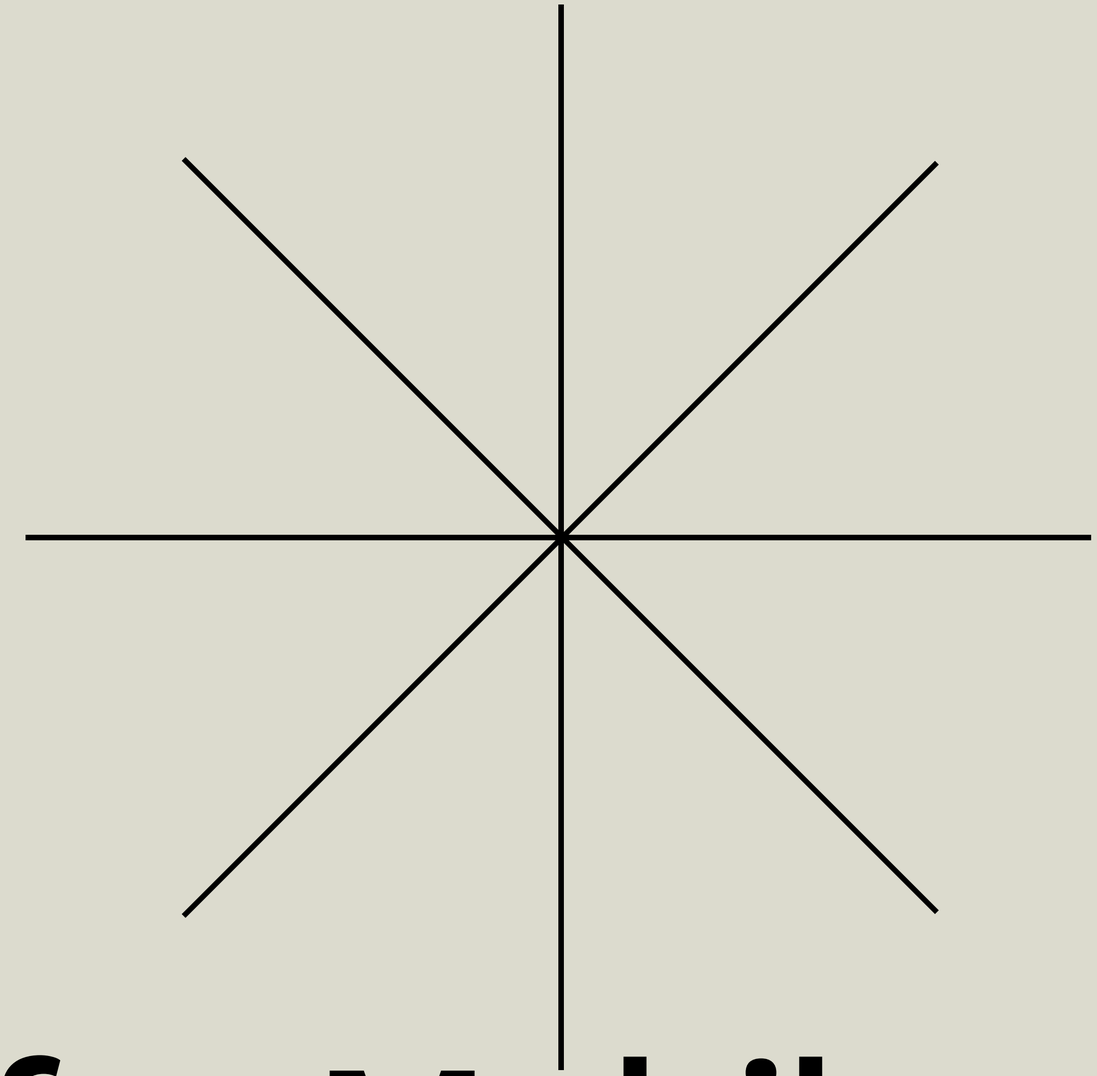
COMANDO FOR

Uma estrutura de repetição que irá executar um código “em loop” até a condição já não se aplicar.

Neste caso, desde que o i seja menor que 5.

```
for (int i = 0; i < 5; i++)
{
    print("Hello!");
}
```

DESENVOLVIMENTO PARA AMBIENTES MÓVEIS



09
Programming for Mobile

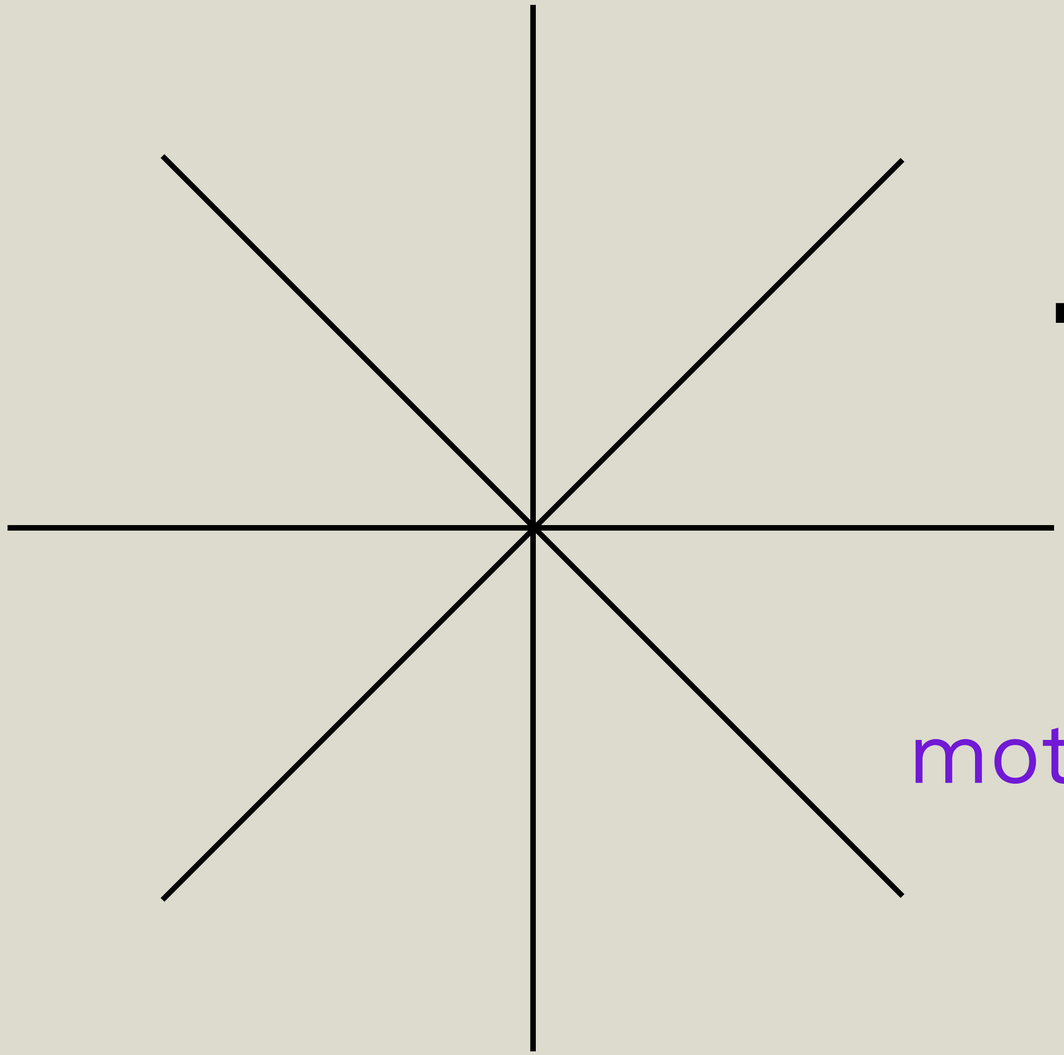
DANIELA MONTEIRO

09. PROGRAMMING FOR MOBILE

In programming for a mobile environment, there are some performance limitations. When developing a video game for a mobile device, the code must be optimized and the assets/materials used must not be heavy.

TOUCH PHASE

Began Moved Stationary Ended Canceled



Thank you!

Don’t forget where to find this powerpoint:

motamdaniela.github.io/dam