BE A CREATOR 1

Learn unity2D

NOVEMBER 24, 2020

EBFIN

Speaking of the author :

Hi, what do you know about the Creator ? about magic ? if you ask me I tell you :

Creator = everyone who creates a new world and story

magic = someone who can make connections

and I tell you the truth , you can be a creator , you can have your story and magic with create a video game.

I say and imagine in your mind:

you have a story , so you need a somewhere to tell it and your story need a animal , people and … you creates all of this , people see your world and play it

Well, I think you just accepted to be a creator , so what you need ? you know?

if you know , this very good , but if you do not know, don't worry I tell you :

1 = You have to choose a game engine , in this book we work with unity2D (I tell you 2d is very good for start)

2 = you have to choose a programing language , for work with unity we us a C# language (it's easy)

3 = you have to find a designer and animation maker (or just do it)

In this book, we work mostly with coding , NO DESIGNER :)

In this book, we creates a 3 different game.

In this book, we see a character system , money management and …

Tanks();

1 = basic C# code

2 = Unity Library

3 = basic create level

4 = basic component in unity

5 = Animation

6 = move

7 = jump

8 = attack

9 = attack 2

10 = pick up

11 = pick up 2

12 = Dialog 1

13 = Dialog 2

Basic C# code:

- ***Types of variables***

We have a different variables , in unity all of components can be a variables, but we have four important variables

1 = **int**

2 = **float**

3 = **string**

4 = **bool**

Int : includes numbers inside the set of integers

for example : …., -1 , 0 , 1 , …

you can't quantify with numbers like : 1.5 , -1.2 , 1.999999 , …

Float : includes all decimal numbers

for example : -1.2, 1.9999, 1.3 , …

String : includes a string of words

For example : "game" , "world" , "games", "1" , …

Bool : includes right or wrong

only can get a two value : true, false

Ok now let's see , how us them :

-I have a question for you , Which variable do you use for money in your game ?

+ ?

- sure , you us int

but let's see how, The general format of the variables is as follows :

(type of variable) (name of variable) = (value)

well, let's see it for int : **int money** = **2000**;

! end of all your lines, you most us this (;)

int -> variable

money -> name for variable

2000 -> value

see for float : **float speed = 1.2f;**

! f means float

float -> variable

speed -> name

1.2 -> value

and you can us this format for all variables

- ***Public vs private***

In unity we need us public and private :

we us them ole variables, but have a lot of different

public : converts a variable to a public one that can be viewed and changed through unity

ex : **public float speed = 1.5f;**

private : converts a variable to a private one that can't be viewed and changed through unity

ex : **private int money = 2000;**

for a speed maybe you need change it from unity but for money you don't need it

- ***Static***

If you have a variable and need to us it in another C# code , you most us static word

ex : **public static float speedCar;**

! if you us static , you can't access variable in unity

all of you code start with a name, for example, Player

for us a static variables in another code you need name code and name variable

ex : **Player.speedCar**

and now you access it

- ***Default functions***

Default functions in unity means , functions No calling in code but they do a work , we have a different functions in unity, ok let's see :

start function : runs only once

update function : runs on a frame basis

about frame , I think you know better than me , but if you don't know about it , I tell you :

think about a very big game , your pc can get 100 fps on it, but my pc can get 60 fps, that means your pc run a update function 100 time but my pc run it 60 time

Awake function : it's like a start function but first run awake code then start code

we us it for a very important things that's most very soon work

FixUpdate function : if your pc get 60 fps , update function run 60 times but fixupdate function run a little more than it

OnEnable : if your object disable when go to enable this function run

OnDisable : of your object enable when go to disable this function run

- ***If, else if , else***

We use these to create condition and compare several things of the same type

! We can only compare variables of the same type!

ex :

```
public int i = 1;
public int b = 1;
public int c = 2;

void start()
{
    if(i == c)
    {
        print("yes")
    }
    else if(i == b)
    {
       print("oh sorry")
    }
    else
    {
        print("no")
    }

}
```

ok , in first if, we check i = c but it's not, so the code inside it does not run

first if don't work so game go to check else if , it's work and print oh sorry in console unity

because the previous condition was correct and fulfilled , so else dose not apply

we see a lot of example in unity code , Please be patient :)

- ***Our functions***

Ok, let's create a function , but first we have to answer the question why we should write a function?

i can tell you 100 reasons, but let's see three of them :

1 = for a beaty

2 = for a button in unity

3 = to build an iterative system

the general form of a function is as follows :

(public or private)(type of function + IEnamrator + void) (name for it) (input ! if have)

{

code

}

Ex :

```
Private int a  = 2;
public void ChangeInt()
{
    a = 0;
}
```

Ex :

```
public string nameMe;
public void AddName(string nameMe)
{
    nameMe = nameMe + "Hi";
}
```

In first function we don't need give input to it , but second function we need

well, next we go to see how called function :

for first type = **ChaneInt();**

for second = **AddName(name) - >** name is a string

! if we don't call our function , never run !

- ***Switch***

It consists of several modes that are executed based on a variable and a condition of one of its modes

- ***Access to components***

Well, in order to make a difference, we also have to create access

Unity Library:

- ***UI***

As the name implies, it is for access to external components , for ex : Image , Text , ...

for import library :

using UnityEngine.UI;

now you can us it, here are a few examples :

```
public Text mymoneyText;

void strat()
{
    mymoneyText.text = "200"
}
```

Description of the above code :

line 1 = this is an object that we can assign text to within Unity

line 3 = it's a function

line 5 = we change a text in unity to our text

```
public Image myImage;
void Start()
```

```
{
    myImage.GetComponent<Image>().color = color.green;
}
```

Description of the above code :

line 1 = this is an object that we can assign image to within Unity

line 2 = it's a function

line 4 = we access the component and change its color

- ***Random***

You can probably guess his work from his name , this library in the simple form that we examine can give us between 2 numbers of luck, Let's see an example together :

```
Public int number;
void start()
{
    number = Random.Rang(0,10);
    print(number);
}
```

Description of the above code :

line1 = it is an integer that is supposed to be chosen by chance

line2 = it's a function

line4 = selects a chance number and places it inside the variable

line5 = prints a number in the output

Build truth and courage? :)