

Projet fin de module

First name : Dahdouh

Last name : Ayoub

CNE : Z206001462

GR : 01

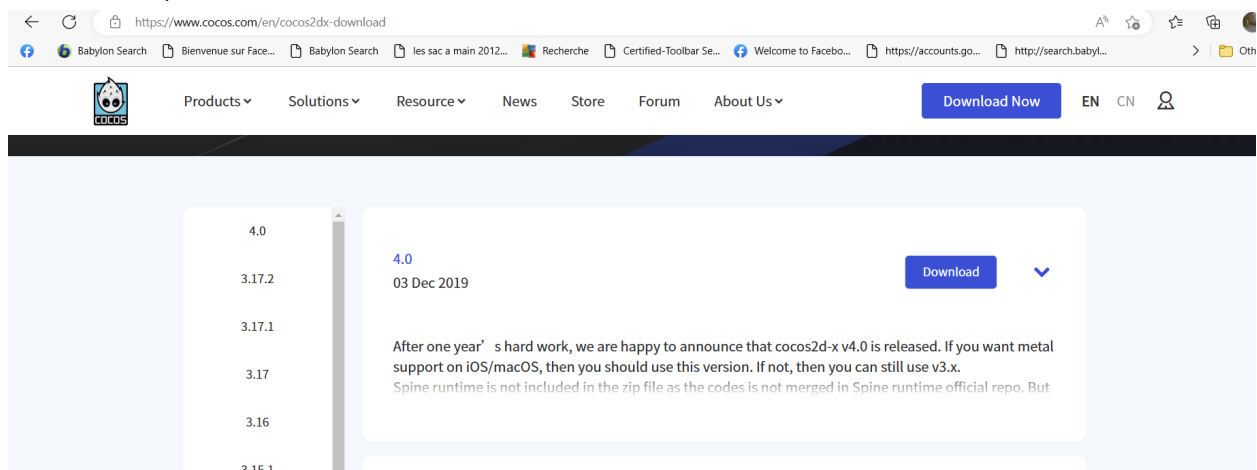
Email : ayoub.dahdouh2@etu.uae.ac.ma

Introduction :

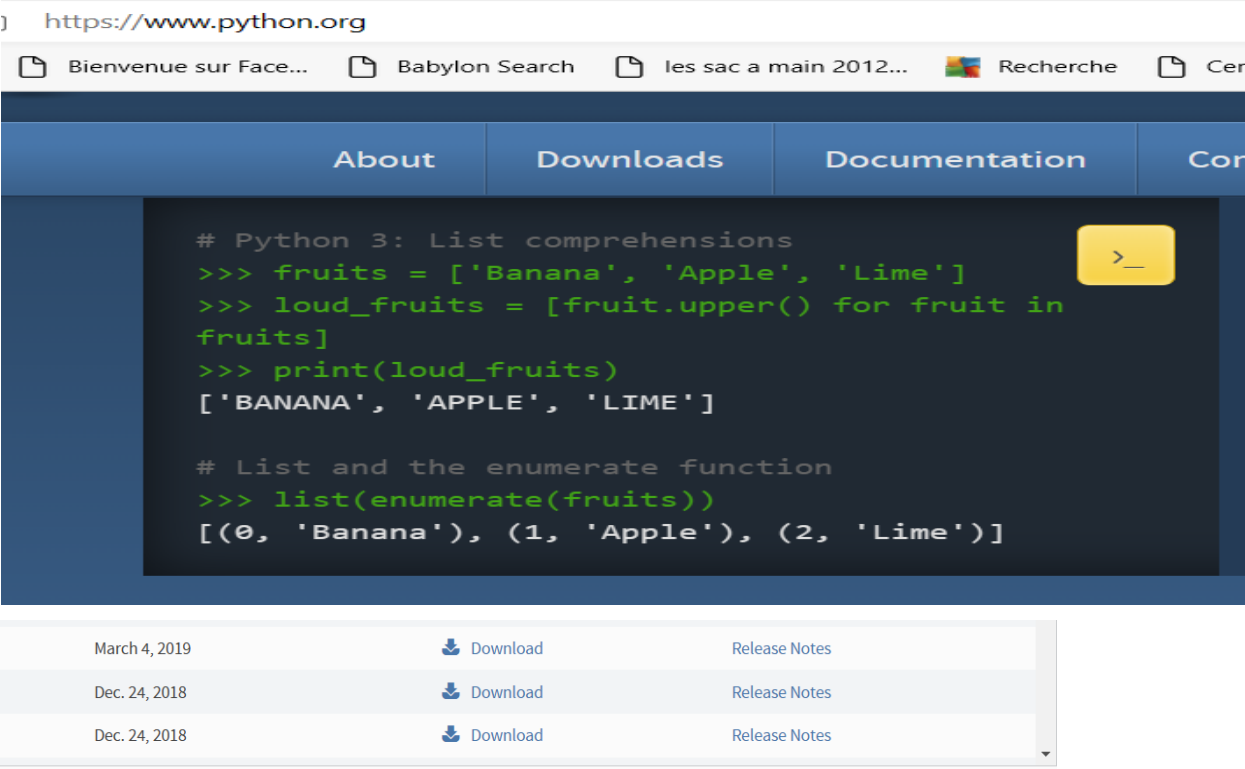
In this project I tried to create a similar picopark game using the cocos2dx engine using cpp, my game contains three levels, each one takes you directly to the next one, I tried to explain the general steps that directed me during this project and the tools that I used and the references too. Also I tried to specify each step and explain it explicitly.

Download and installation :

1) Cocos download & installation :



2) Python download & installation :



The screenshot shows the Python.org website. The top navigation bar includes links for About, Downloads, Documentation, and Contribute. Below the navigation bar, there is a code editor with the following Python code:

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Below the code editor, there is a table listing the latest versions of Python for Windows:

Python Version	Release Date	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes
Python 3.6.8	Dec. 24, 2018	Download	Release Notes

Below the table, there is a screenshot of a Windows Command Prompt window titled "Command Prompt - python". The window shows the output of the command "python" in a Windows 10 environment:

```
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ayoub>python
Python 2.7.16 (v2.7.16:413a49145e, Mar  4 2019, 01:37:19) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

3) Cmake download & Installation :

```

C:\Users\ayoub>cmake
Usage

  cmake [options] <path-to-source>
  cmake [options] <path-to-existing-build>
  cmake [options] -S <path-to-source> -B <path-to-build>

Specify a source directory to (re-)generate a build system for it in the
current working directory. Specify an existing build directory to
re-generate its build system.

Run 'cmake --help' for more information.

C:\Users\ayoub>

```

Platform	Files
Windows x64 Installer:	cmake-3.25.1-windows-x86_64.msi
Windows x64 ZIP	cmake-3.25.1-windows-x86_64.zip

4) Extract cocos :

Extracting cocos took a very much long while which I hasn't taken any screenshot of it unfortunately but we will the result in the command prompt

```

C:\Users\ayoub>cmake
Usage

  cmake [options] <path-to-source>
  cmake [options] <path-to-existing-build>
  cmake [options] -S <path-to-source> -B <path-to-build>

Specify a source directory to (re-)generate a build system for it in the
current working directory. Specify an existing build directory to
re-generate its build system.

Run 'cmake --help' for more information.

C:\Users\ayoub>

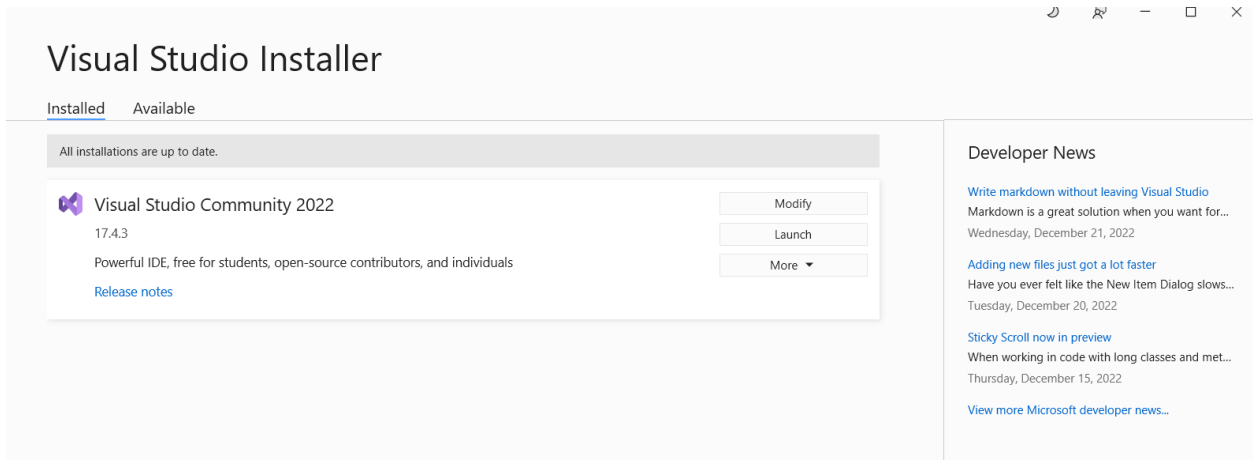
```

Creating a project :

```
Command Prompt
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ayoub>cocos new actualgame -l cpp -p com.sonarsystems.actualgame
```

Visual Studio Community 2022 download and installation :



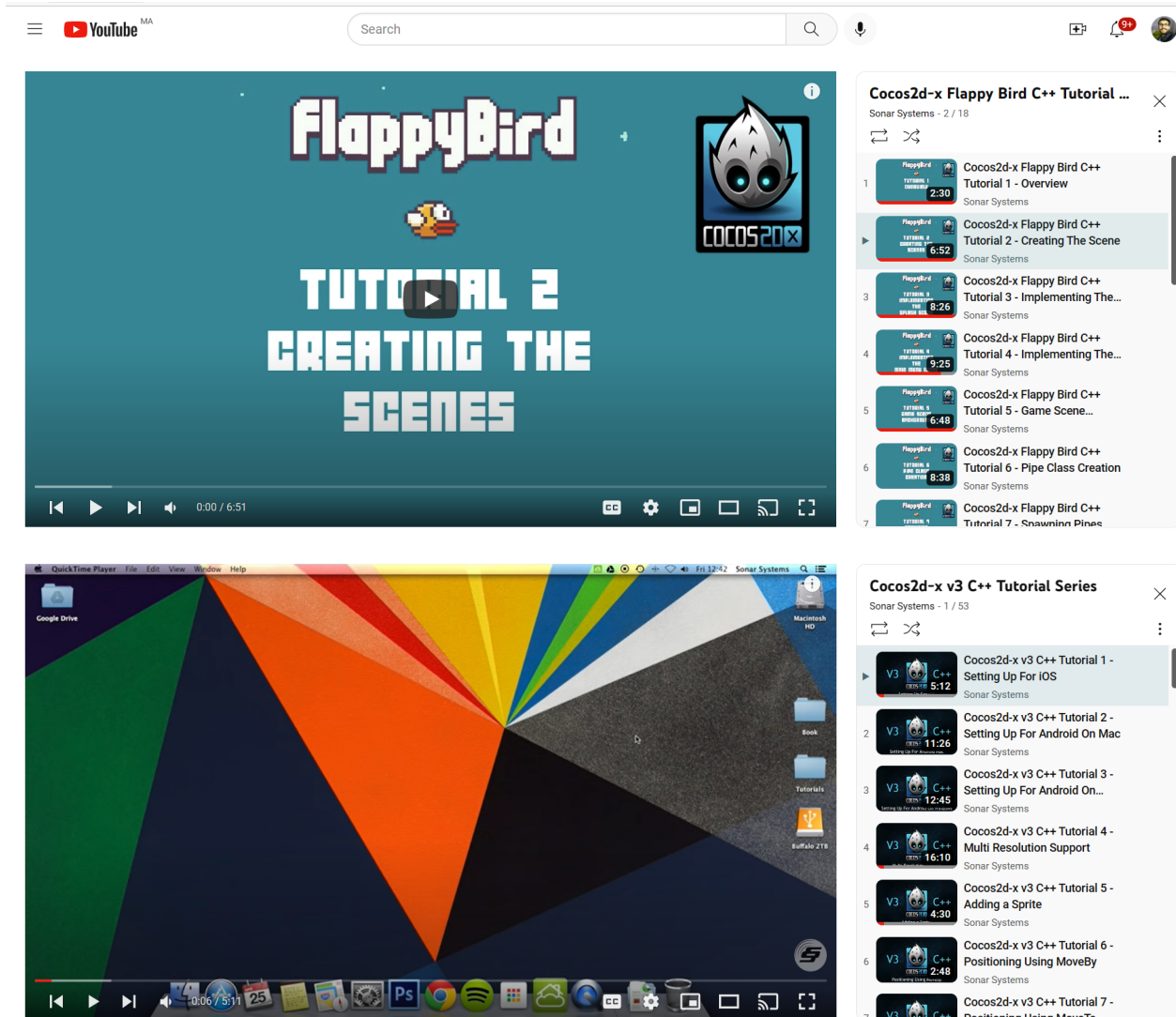
Create a visual studio project :

We move the projwin32 of our new project to the command prompt and then we type the next command : `cmake .. -G "Visual Studio 17 2022" -A Win32`























Afterwards we go to our project and we open the file " actualgame.sln which leads us directly to our project in visual studio .








Creating the project :

First of all I searched for videos in youtube that could help me to get familiar with the cocos coding, which I have found multiple very useful ones



Also I searched for game codes with cocos2dx cpp in github and I luckily found various ones which have helped me a lot to make this game work as it does. So I cloned them into my computer using the command clone "git link"

<div> <div> <div>←</div> <div>→</div> <div>⬆</div> <div>⬇</div> <div>⬇</div> <div>⬆</div> </div> <div> <div>« OS (C:) »</div> <div>Users »</div> <div>ayoub »</div> <div>Desktop »</div> <div>RollerSplat</div> </div> <div> <div>↺</div> <div>↻</div> </div> <div> <div>🔍</div> <div>Search RollerSplat</div> </div> </div>				
Name	Date modified	Type	Size	
 AppDelegate	12/11/2022 0:43	C++ Source File	5 KB	
 AppDelegate	12/11/2022 0:43	C Header File	3 KB	
 ball	12/11/2022 0:43	PNG File	7 KB	
 ballShadow	12/11/2022 0:43	PNG File	12 KB	
 button	12/11/2022 0:43	PNG File	19 KB	
 cadre	12/11/2022 0:43	C++ Source File	3 KB	
 cadre	12/11/2022 0:43	C Header File	1 KB	
 cadre	12/11/2022 0:43	PNG File	41 KB	
 form	12/11/2022 0:43	PNG File	6 KB	
 formbot	12/11/2022 0:43	PNG File	24 KB	
 formright	12/11/2022 0:43	PNG File	28 KB	
 formtop	12/11/2022 0:43	PNG File	23 KB	
 GameOver	12/11/2022 0:43	C++ Source File	2 KB	
 GameOver	12/11/2022 0:43	C Header File	1 KB	
 HelloWorld	12/11/2022 0:43	PNG File	37 KB	
 level1	12/11/2022 0:43	C++ Source File	13 KB	
 Level1Scene	12/11/2022 0:43	C++ Source File	13 KB	
 Level1Scene	12/11/2022 0:43	C Header File	2 KB	
 Level2Scene	12/11/2022 0:43	C++ Source File	14 KB	
 Level2Scene	12/11/2022 0:43	C Header File	1 KB	
 Level3scene	12/11/2022 0:43	C++ Source File	17 KB	
 Level3scene	12/11/2022 0:43	C Header File	1 KB	
49 items				

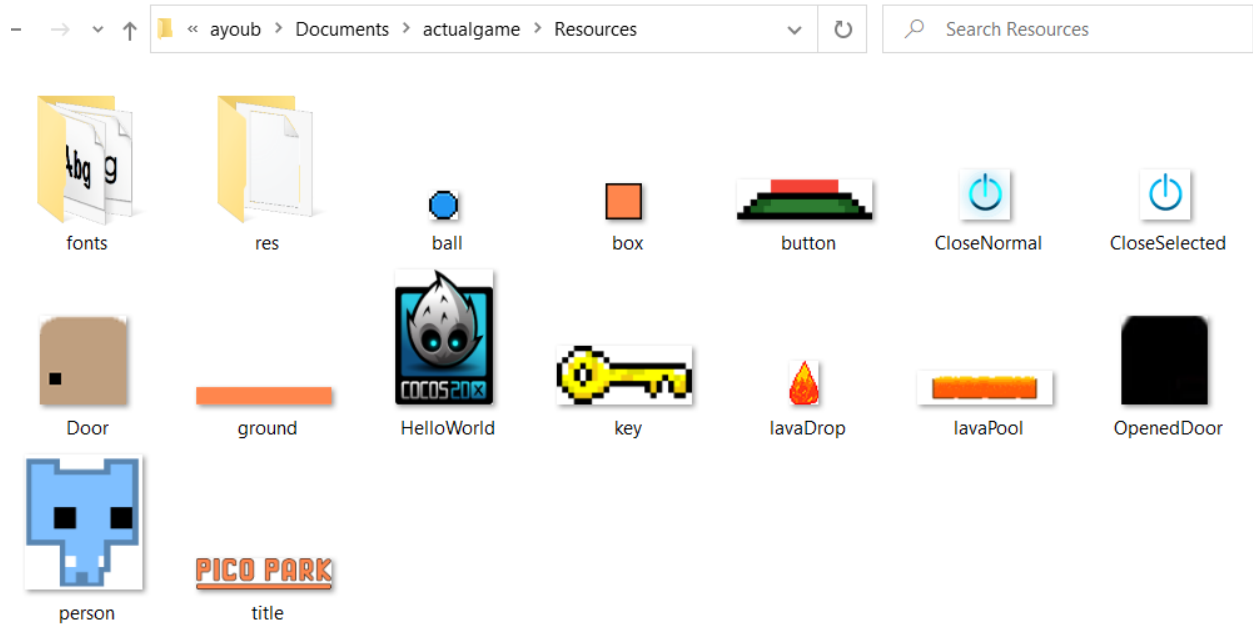
<div> <div> <div>←</div> <div>→</div> <div>⬆</div> <div>⬇</div> <div>⬇</div> <div>⬆</div> </div> <div> <div>> This PC »</div> <div>Downloads »</div> <div>cocos2d-x-samples »</div> <div>samples</div> </div> <div> <div>↺</div> <div>↻</div> </div> <div> <div>🔍</div> <div>Search samples</div> </div> </div>				
Name	Date modified	Type	Size	
 FantasyWarrior3D	12/12/2022 13:16	File folder		
 GAFExamples	12/12/2022 13:16	File folder		
 KillBug	12/12/2022 13:16	File folder		
 LiquidFun-EyeCandy	12/12/2022 13:16	File folder		
 LiquidFun-Testbed	12/12/2022 13:16	File folder		
 MoonWarriors	12/12/2022 13:16	File folder		
 SwiftTetris	12/12/2022 13:16	File folder		

These codes helped me a lot creating this game as it which was a very difficult task .

The project :

First I searched for the pictures that I will need and I transformed then into png


And I put them in resources

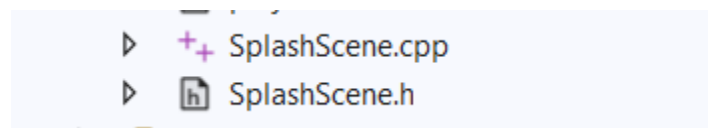


6 items



Then I started following the videos and the samples instructions part by part !
So I created a new class named Spashscene that controls the background appearance
Which shows the game title for a few seconds

 SplashScene	1/5/2023 19:35	C++ Source File	2 KB
 SplashScene	12/19/2022 16:22	C Header File	1 KB

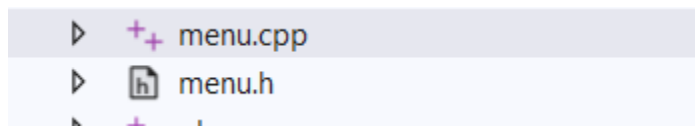


```

1  #include "SplashScene.h"
2  #include "menu.h"
3
4  USING_NS_CC;
5
6  Scene* SplashScene::createScene()
7  {
8      //Creating a scene
9      auto scene = Scene::create();
10     //Creating a layer
11     auto Layer = SplashScene::create();
12     //adding layer to scene
13     scene->addChild(Layer);
14     //returning the scene created
15     return scene;
16 }
17
18 // Print useful error message instead of segfaulting when files are not there.
19 static void problemLoading(const char* filename)
20 {
21     printf("Error while loading: %s\n", filename);
22     printf("Depending on how you compiled you might have to add 'Resources/' in front of filenames in HelloWorldScene.cpp\n");
23 }
24
25 // on "init" you need to initialize your instance
26 bool SplashScene::init()
27 {
28     // 1. super init first
29     if ( !Scene::init() )
30     {
31         return false;
32     }
33     LayerColor* _bgColor = LayerColor::create(Color4B::WHITE);

```

Then I created a new class named Menu which shows a clickable button “ Play “ that leads you to the first level of the game

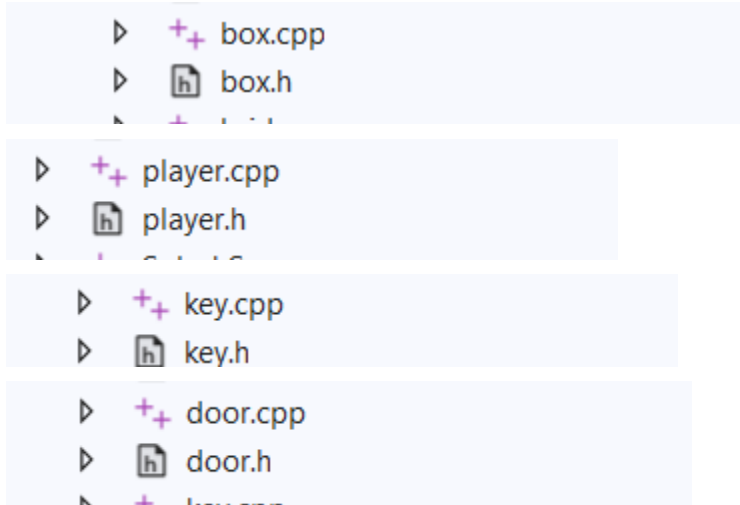


```

1  #include "menu.h"
2  #include "level1.h"
3
4
5
6  USING_NS_CC;
7
8  Scene* MainMenu::createScene()
9  {
10     // 'scene' is an autorelease object
11     auto scene = Scene::create();
12
13     // 'layer' is an autorelease object
14     auto layer = MainMenu::create();
15
16     // add layer as a child to scene
17     scene->addChild(layer);
18
19     // return the scene
20     return scene;
21 }
22
23 // on "init" you need to initialize your instance
24 bool MainMenu::init()
25 {
26     ///////////////////////////////////
27     // 1. super init first
28     if (!Layer::init())
29     {
30         return false;
31     }
32     LayerColor* _bgColor = LayerColor::create(Color4B(255, 255, 255, 255));
33     // add background color

```

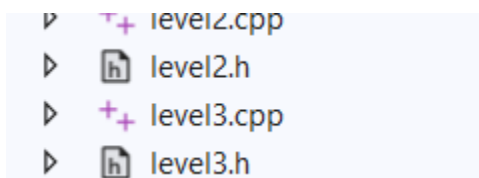
Afterwards simultaneously I created the classes of the player and the box class that controls the background image and the door class and the key class it's shown in the code

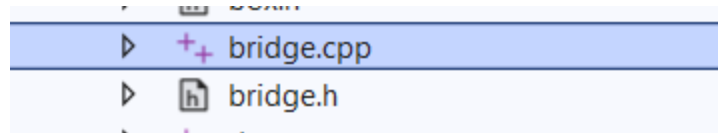


And then I started creating my first level , simultaneously modifying the previous codes when it's necessary



And then Level 2 and 3 alongside with them we will find a class named bridge that we will need in the level two





```
bridge::bridge(void) {
    this->isClaimed = false;
};

bridge* bridge::bridgeinit() {
    auto sprite = new bridge();
    sprite->initWithFile("box.png");
    sprite->setScale(1.2);
    return sprite;
}
```

```
1  #include "level2.h"
2  #include "ui/UITextField.h"
3  #include "level3.h"
4
5  USING_NS_CC;
6  Scene* Lvl2::createScene()
7  {
8      auto scene = Scene::createWithPhysics();
9      scene->getPhysicsWorld()->setGravity(Vec2(0, -400));
10     auto layer = Lvl2::create();
11     scene->addChild(layer);
12     return scene;
13 }
14
15 // Print useful error message instead of segfaulting when files are not there
16
17 // on "init" you need to initialize your instance
18 bool Lvl2::init()
19 {
20     if (!Layer::init())
21     {
22         return false;
23     }
24
25     //-----create backgroud color-----
26
27     LayerColor* _bgColor = LayerColor::create(Color4B::WHITE);
28     this->addChild(_bgColor, -10);
29
30     //-----create variables for positioning our instance-----
31
32     auto visibleSize = Director::getInstance()->getWinSize();
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

No issues found

Ln: 3 Ch: 1 SPC M

The source code explain every step of the game

```
#include "Level3.h"
#include "menu.h"
#include "ui/UITextField.h"

USING_NS_CC;

Scene* Lvl3::createScene()
{
    auto scene = Scene::createWithPhysics();
    scene->getPhysicsWorld()->setGravity(Vec2(0, -400));
    //scene->getPhysicsWorld()->setDebugDrawMask(0xffff);
    auto layer = Lvl3::create();
    scene->addChild(layer);
    return scene;
}

// Print useful error message instead of segfaulting when files are not there

// on "init" you need to initialize your instance
bool Lvl3::init()
{
    if (!Layer::init())
    {
        return false;
    }

    //-----create background color-----

    LayerColor* _bgColor = LayerColor::create(Color4B::WHITE);
    this->addChild(_bgColor, -10);

    // create variables for positioning our instance
}
```

I faced some problems while debugging the code but I installed msvcrt110.dll and carried it in the system32 (version 64bits) AND the version 32bits in sysWOW64 and then my code worked fine.

Conclusion :

So, the game has three levels each level lead u to the next and when u fail in a level , that level get updated until you pass it successfully

As a resume, working with cocos-2dx was very hard actually, finding the right information wasn't easy and I'm still learning tons of things about it, because it uses a lot of libraries and functions that I have never seen in c++ so far .

Last but not least, I hope you enjoy the game!