

C++ training result

Nguyen Anh Phuong



Programming Theory

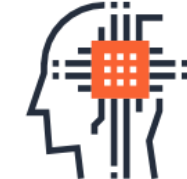
Basic game Theory

Development Orientation



Programming Theory

- Pointer
- Function Pointer
- String C Type (Char *)
- Array
- Function Template
- Heap & Stack
- Linked List



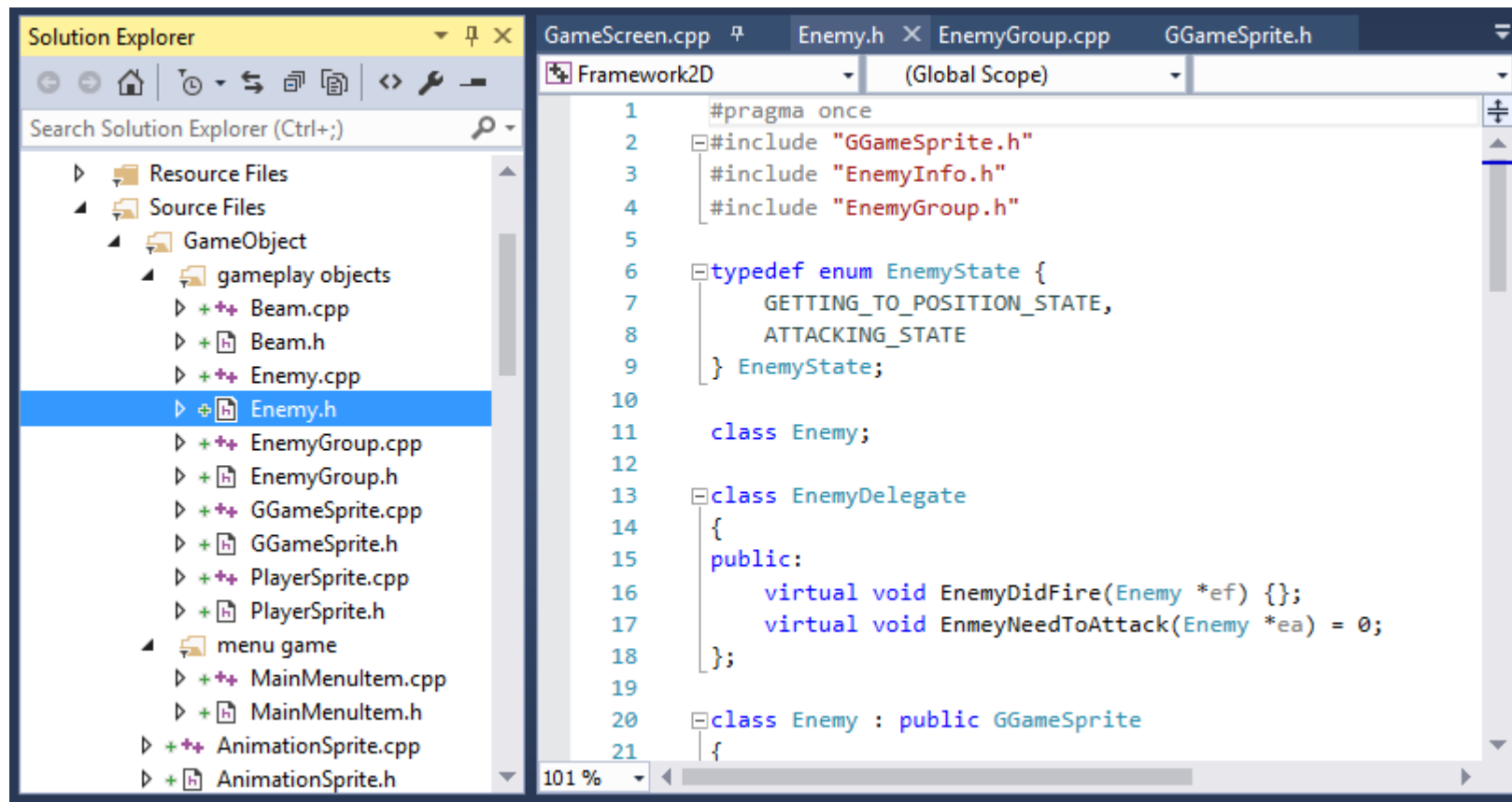
Basic Game Theory

- Game Object (OOP)
- Delegate Pattern
- Game Resources
- Game Resource Manager
- Current Features



Basic Game Theory

Game Object (OOP)



Basic Game Theory

Delegate Pattern

Call back to the delegate object.
(protocol in Objective-C)

```
GameplayScreen.h
Framework2D
GameplayScreen

18
19 class GameplayScreen : public GameScreen
20     , public PlayerSpriteDelegate
21     , public EnemyDelegate
22     , public InGameMenuDelegate
23 {
24 protected:
25     bool init(int stage);
26
27     void PlayerSpriteDidFire(PlayerSprite *ps);
28     void PlayerSpriteDidLevelUp(PlayerSprite *ps);
29     void PlayerSpriteDidChangeHP(PlayerSprite *ps);
30     void EnemyDidFire(Enemy *ef);
31     void EnemyNeedToAttack(Enemy *ea);
32     void InGameMenuDidConfirm(InGameMenuItemType type);
33
34 public:
35     GameplayScreen();
```

```
Enemy.h
Framework2D
EnemyState

1 #pragma once
2 #include "GGameSprite.h"
3 #include "EnemyInfo.h"
4 #include "EnemyGroup.h"
5
6 typedef enum EnemyState {
7     GETTING_TO_POSITION_STATE,
8     ATTACKING_STATE
9 } EnemyState;
10
11 class Enemy;
12
13 class EnemyDelegate
14 {
15 public:
16     virtual void EnemyDidFire(Enemy *ef) {};
17     virtual void EnemyNeedToAttack(Enemy *ea) = 0;
18 };
19
20 class Enemy : public GGameSprite
21 {
22 public:
23     static Enemy *Create(int level, bool isBoss = false);
```

Basic Game Theory

Game resources



ork2D > Resources > Data

Name

config.json
Enemy.json
enemyGroup.json
playerLevel.json
stages.json

D > Resources > Textures



Search Textures



A1.png



A2.png



A3.png



A4.png



A5.png



A6.png



A7.png



A8.png



A9.png



A10.png



A11.png



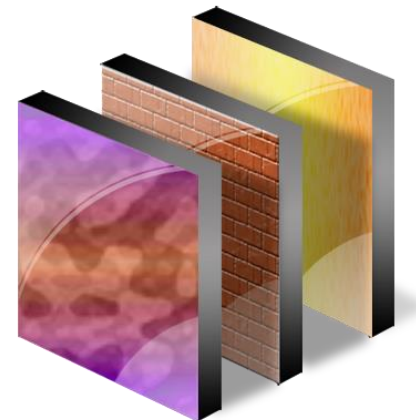
A12.png



Basic Game Theory

Game resource manager

- Singleton Pattern
- Read/Write .json file
- Parse & Cache .json information to objects
- Cache texture Images



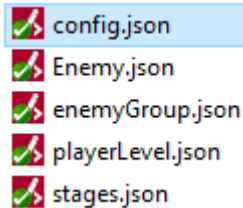
Basic Game Theory

Current Features

- Menu UI.
- Implement gameplay state.
- Save/Load game.
- Sprite Animation.
- Balance & Config Game - json data files.



Balancing & Configuration



Settings - config.json

```
{
  "screenWidth":800,
  "screenHeight":600,
  "FPS":60,
  "playerMoveSpeed":300,
  "beamSpeed":500,
  "enemyStopChasingY":150,
  "startStagePlayerHP":10,
  "playerSize":70,
  "enemySize":60,
  "bossSize":5,
  "bossHP":20,
  "bossAtkSpd":4
}
```

```
[
  {
    "level": 1,
    "hp": 10,
    "ms": 200,
    "as": 10,
    "res": [
      "A1.png",
      "B1.png",
      "C1.png",
      "D1.png",
      "E1.png",
      "F1.png",
      "G1.png"
    ]
  },
  {
    "level": 2,
    "hp": 15,
    "ms": 200,
    "as": 9,
    "res": [
      "A2.png",
      "B2.png",
      "C2.png",
      "D2.png",
      "E2.png",
      "F2.png",
      "G2.png"
    ]
  }
]
```

```
[
  {
    "level": 1,
    "beam type": 1,
    "beam damage": 10,
    "score to next": 100
  },
  {
    "level": 2,
    "beam type": 2,
    "beam damage": 20,
    "score to next": 200
  }
]
```

```
[
  {
    "stage": 1,
    "enemy count": 10,
    "enemy spawn time": 0.3,
    "enemy min level": 1,
    "enemy max level": 1
  },
  {
    "stage": 2,
    "enemy count": 11,
    "enemy spawn time": 0.4,
    "enemy min level": 2,
    "enemy max level": 2
  }
]
```

Development Orientation

- Texture pack.
- Texture Sprite Animation.
- Particle systems.
- Game sound.
- Enhancing enemy AI.
- Game contents (items, upgrading, bullet type...)
- Online Multiplayer

