

# 学习计划

---

以前试过不少计划表，但最后效果都不是很好。然后慢慢就习惯了只制定截止日期，让那些精确到时分秒的计划表都去死吧。平时习惯把要做的事都列在印象笔记里，不过写得太随便就不截图了。

这次的自学计划大概是这样的：以27号为界完成大部分预定任务，因为之后还要去拜年；需要完成的任务大概是这样的：

- 作业中的视频课程；
- 通过做题复习算法和数据结构；
- C#入门。。
- 整理现在遇到过的游戏功能，并优化拓展；
- 在博客整理归纳一些编程的小技巧和常遇到的问题。
- 习惯一些“编程规范”。

视频课程现在学到了课时24，大概每天两三个课时，有时多一些，时断时续；

刷题的话，平时爱用codevs和vijos。本来准备继续做PAT的题，但是甲级好像不是很难，顶级还没看，主要还是英文题目阅读起来比较困难，还要锻炼锻炼。以前用Pascal爬过codevs的天梯，现在正立志于用C++再爬一次，好艰辛。。上学期就时不时做了一些，现在陷于动规的泥潭中；

关于C#。。就那样吧反正Unity够用就好；

整理游戏功能我是准备放在最后做的，现在还没开始；

乱七八糟的小技巧正陆续完成，主要是想写一下关于sort的各种用法；

“编程规范”是我开始学Unity之后才重视起来的，因为工程代码是清晰第一，效率第二，它需要易于理解和维护，按照统一的标准来编写。在写算法题时，我还是认为简单是最好的，不用累赘的函数、一大串函数名和乱七八糟的变量名，效率优先，不只是运行效率。而对于现在还在整理的游戏代码，都尽量采用规范的方式编写，虽然还是觉得别扭，但也慢慢习惯了。

关于书的话，除了C Primer Plus，也可以看看[C程序设计语言](#)，这是开发出C语言的两位大佬Ken Thompson和Dennis Ritchie写的，配合配套的习题集食用更佳。毕竟，谭大大的那本书错误还是蛮多的，内容也有些老了。书都挺好找的，网上PDF漫天飞，学校图书馆好像也有。不过，学算法和数据结构就另当别论了。

编程是为了解决问题，而不是为了炫技，还是希望自己能写出更漂亮的程序。

最后附上一些整理过的代码，也算是自查的结果了。

初学时写的一个人物移动代码

```
4 public class move : MonoBehaviour
5 {
6     public Rigidbody2D x;
7     public float moves;
8     public float jumps;
9     private bool jump;
10
11     public Transform groundc;
12     public float grounds;
13     public LayerMask ground;
14     private bool onground;
15
16     private float dir;
17
18     void Start()
19     {
20         x = GetComponent<Rigidbody2D>();
21     }
22
23     void Update()
24     {
25         onground = Physics2D.OverlapCircle(groundc.position, grounds, ground);
26         if (Input.GetKey(KeyCode.LeftArrow))
27         {
28             x.velocity = new Vector2(-moves, x.velocity.y);
29             dir = 1;
30         }
31         if (Input.GetKey(KeyCode.RightArrow))
32         {
33             x.velocity = new Vector2(moves, x.velocity.y);
34             dir = -1;
35         }
36         if (Input.GetKey(KeyCode.UpArrow))
37         if (onground) x.velocity = new Vector2(x.velocity.x, jumps);
38         if (jump)
39         {
40             if (onground) x.velocity = new Vector2(x.velocity.x, jumps);
41             jump = false;
42         }
43         GetComponent<Animator>().SetFloat("dir", dir);
44     }
45 }
```

寒假学着一个Asset写的相机代码

```
1 using UnityEngine;
2
3 public class CameraFollow : MonoBehaviour
4 {
5     public Controller2D target;
6     public float verticalOffset;
7     public float lookAheadDstX;
8     public float lookSmoothTimeX;
9     public float verticalSmoothTime;
10    public Vector2 focusAreaSize;
11
12    private FocusArea focusArea;
13
14    private float currentLookAheadX;
15    private float targetLookAheadX;
16    private float lookAheadDirX;
17    private float smoothLookVelocityX;
18    private float smoothVelocityY;
19
20    private bool lookAheadStopped;
21
22    private void Start()
23    {
24        focusArea = new FocusArea(target.coll.bounds, focusAreaSize);
25    }
26
27    private void LateUpdate()
28    {
29        focusArea.Update(target.coll.bounds);
30
31        Vector2 focusPosition = focusArea.centre + Vector2.up * verticalOffset;
32
33        if (focusArea.velocity.x != 0)
34        {
35            lookAheadDirX = Mathf.Sign(focusArea.velocity.x);
36            if (Mathf.Sign(target.playerInput.x) == Mathf.Sign(focusArea.velocity.x) && target.playerInput.x != 0)
37            {
38                lookAheadStopped = false;
39                targetLookAheadX = lookAheadDirX * lookAheadDstX;
40            }
41            else
```