

代码

周潇翔

摘要. 这里总结自己学过的代码供查阅。为啥不用英文？英文的参考文献浩如烟海，也不差我一个啊。另外引用的很多网页也都是中文的，我也懒得再花时间解释哪个是中文哪个是英文了。对数学系的同学而言，代码的逻辑并不难，大家不会的只是格式而已。第一节横向介绍我们需要啥，之后纵向对每一种语言给出对应的代码。

在这份文档的编译中第一次学 python 和 Github, 可以说是紧跟潮流
23333

1. 代码需求

大部分的语言都需要：

- 安装 + 初始代码 (Halloworld)
- 基本逻辑
- 调试
- 参考文档
- (想保留的) 例子

以下是具体需求：

1.1. 安装 + 初始代码.

- 简要说明该语言的目的
- 说明自己使用何种编译器
- 解释该语言的结构 (基本框架)
- 使用该语言在屏幕中打出”Halloworld”
- 必要时给出英文注释

1.2. 基本逻辑.

- 数据结构类型 (数字、字符串、其他结构)
- 基本四则运算 + mod (若数据结构中包含矩阵，则需要矩阵的各类运算；)
- 条件语句
- 循环语句

- 函数

对于图形化输出的语言，我们还有很多额外的需求：

- 图形

- 基本几何图形 (尽可能多样化)
- Bézier 曲线 (钢笔工具)
- 参数化曲线
- 直线，箭头等基本图形库，电路，棋盘，乐器等专业库
- 几何变换 (平移，旋转，伸缩)
- 集合运算 (如交并补)
- 几何运算 (如平行，垂直，切线，交点)
- 度量衡转换 (数值，距离单位，向量运算，弧度制与角度制)
- 测量

- 图形进阶

- 光滑化
- 锐化

- 坐标系

- 直角坐标系
- 其他类型坐标系

- 其他细节

- 颜色设置
- 背景与边界设置
- 透明背景设置
- 图片嵌入设置
- 材质设置
- 数学字体嵌入设置
- 动画 + 声音嵌入设置 (可选)

对于主要用于排版的语言，我们还希望能了解页面布局的相关知识，如盒子模型。

1.3. 调试.

- 快捷键 (Keyboard shortcut)

- 运行代码
- 注释方式及快捷键 (单行注释 + 多行注释)
- 自动补全功能
- 自动对齐功能

– 其他快捷键

- 如何获得帮助
- 控制输出 (包含调整至 debug 模式)
- 如何设置断点
- 控制输入

1.4. 参考文档.

- 官方文档
- 民间优秀文档

超出科大 C 语言的知识: 编程范式 (Programming paradigm)、方法 (method)

2. PYTHON

2.1. 安装 + 初始代码. Python 是一门高级的编程语言。他有许多的标准模块 (standard module).

2.2. 基本逻辑. 与 C 语言不同, Python 不需要声明变量。当有赋值时不输出结果。

数据类型在这里看到。¹

对于数字, Python 不仅有 int 和 float, 还有 Decimal, Fraction and complex numbers 这些奇葩的变量。Task: 学会 Decimal, Fraction.

```
1 >>> complex('1+2j')*complex('1+3j')
```

四则计算像自然计算一样自然, 不过带余除法用//, 余数用%, 幂次用**.(好符号)²

```
>>> a,b=8,13 # a++ is not allowed in python
>>> a ** (b-1) % b # verify the Fermat's little theorem
```

计算器上的 Ans 记为_, round(0.142857,1)给出 0.1

python 中的逻辑运算符如下: 与 (and &), 或 (or |)

条件语句和循环语句的书写规范详见这里。以下是计算素数的例子。

```
import math    # Compute square root
def isPrime(n):    # return true when n is a prime
    for x in range(2, math.isqrt(n)+1):
        if n % x == 0:
5             return False;
```

¹你需要知道那些类型是可改变的 (mutable);

²请小心使用负数的带余除法。

```

    else:
        # loop fell through without finding a factor
        return True;
isPrime(57)

```

用库可以更加容易地计算素数:

```

1 from sympy.ntheory import isprime
  isprime(10000019)

# find primes by sieve method
from sympy import sieve
3 sieve._reset() # this line for doctest only
  # 10000019 in sieve    #10000019 is a prime
  sieve.extend(100)
  sieve._list

```

2.3. 调试. 单行注释使用井号#, 多行注释使用'''注释'''or """注释""". 快捷键为 Ctrl+./.

控制输入使用函数input()。

2.4. 参考文档. 官方文档 SymPy PKU CS 自学指南

2.5. (想保留的) 例子. 这个例子计算正整数的各位数之和, 用到了把数转化为字符串的技巧:

```
print(sum([int(d) for d in str(int(input("number:"))])))
```

这个例子计算 gl_n 幂零轨道的维数.

```

# this computes dimension of orbits of nilpotent elements
young_diagram = [2,1,1,1,1] # Here is the input
n=len(young_diagram)
4 a,b=0,0
  for i in range(n):
    a=young_diagram[i]
    for j in range(n):
      if young_diagram[i]<young_diagram[j]:
9         b=young_diagram[i]
      else:
        b=young_diagram[j]
c=a**2 -b
print("the dimension of this orbit is", str(c)+".") # one small trick for
printing result

```

这个例子想要验证某个猜想。据说是 BSD 猜想的推论。

```
from sympy.ntheory import isprime
```

```

2 import math      # Compute square root
def isQuart(k):
    for i in range(k//2):
        if i**4 % k == 3:
            return True
    return False
7
def solution(m):
    n=math.isqrt(m)+1
    num=0
    for i in range(n):
12         for j in range(n):
            for k in range(n):
                if 6* i**2+j**2+18* k**2 == m:
                    #print("find the solution", i, j, k)
                    if i*j*k==0:
17                         if k % 2 ==0:
                            num=num+4
                        else:
                            num=num-4
                    else:
22                         if k % 2 ==0:
                            num=num+8
                        else:
                            num=num-8
    return num
27 for l in range(25, 10000, 24):
    if isprime(l) & (isQuart(l)==False):
        #print(l,"find the number")
        if solution(l) % 16 ==8:
            print("the conjecture is true for ", l)
32     else:
        print("the conjecture is not true for ", l)

```

这个例子计算 affine quiver 正根的可能情形 (real positive root + regular simple).

```

# Usage of format
2 # E6 case
orderedCouple1 = [(a1,a2,a3,a4,a5,a6,a7) for a1 in range(2) for a2 in
    range(3) for a3 in range(4) for a4 in range(3) for a5 in range(2) for
    a6 in range(3) for a7 in range(2)]
k=0 # compute the number of positive real root which is possible regular
    simple
for a1,a2,a3,a4,a5,a6,a7 in orderedCouple1: # do everything in one line*

```

```

if a1**2+a2**2+a3**2+a4**2+a5**2+a6**2+a7**2-a1*a2-a2*a3-a3*a4-a4*a5-
a3*a6-a6*a7 ==1:
7     k=k+1
    print('                                {0}'.format(a7)) #ugly code
    have better output
    print('                                {0}'.format(a6))
    print('The positive real root is {0}, {1}, {2}, {3}, {4}.'.format(
a1,a2,a3,a4,a5))
    print("")
12 print('There are {0} results'.format(k))

# E7 case
orderedCouple2 = [(a1,a2,a3,a4,a5,a6,a7,a8) for a1 in range(2) for a2 in
range(3) for a3 in range(4) for a4 in range(5) for a5 in range(4) for
a6 in range(3) for a7 in range(2) for a8 in range(3)]
k=0 # compute the number of positive real root which is possible regular
simple
17 for a1,a2,a3,a4,a5,a6,a7,a8 in orderedCouple2: # do everything in one line
*
    if a1**2+a2**2+a3**2+a4**2+a5**2+a6**2+a7**2+a8**2-a1*a2-a2*a3-a3*a4-
a4*a5-a5*a6-a6*a7-a4*a8 ==1:
        k=k+1
        print('                                {0}'.format(a8)) #ugly
code have better output
        print('The positive real root is {0}, {1}, {2}, {3}, {4}, {5},
{6}.'.format(a1,a2,a3,a4,a5,a6,a7))
22        print("")
    print('There are {0} results'.format(k))

# E8 case
orderedCouple3 = [(a1,a2,a3,a4,a5,a6,a7,a8,a9) for a1 in range(2) for a2
in range(3) for a3 in range(4) for a4 in range(5) for a5 in range(6)
for a6 in range(7) for a7 in range(5) for a8 in range(3) for a9 in
range(4)]
27 k=0 # compute the number of positive real root which is possible regular
simple
for a1,a2,a3,a4,a5,a6,a7,a8,a9 in orderedCouple3: # do everything in one
line*
    if a1**2+a2**2+a3**2+a4**2+a5**2+a6**2+a7**2+a8**2+a9**2-a1*a2-a2*a3-
a3*a4-a4*a5-a5*a6-a6*a7-a7*a8-a6*a9 ==1:
        k=k+1
        print('                                {0}'.format(a9)) #
ugly code have better output
32        print('The positive real root is {0}, {1}, {2}, {3}, {4}, {5},
{6}, {7}.'.format(a1,a2,a3,a4,a5,a6,a7,a8))

```

```

        print("")
    print('There are {0} results'.format(k))

    # E6 case, subspace case
37 orderedCouple1 = [(a1,a2,a3,a4,a5,a6,a7) for a1 in range(2) for a2 in
    range(3) for a3 in range(4) for a4 in range(3) for a5 in range(2) for
    a6 in range(3) for a7 in range(2)]
    k=0 # compute the number of positive real root which is possible regular
    simple
    for a1,a2,a3,a4,a5,a6,a7 in orderedCouple1: # do everything in one line*
        if a1**2+a2**2+a3**2+a4**2+a5**2+a6**2+a7**2-a1*a2-a2*a3-a3*a4-a4*a5-
            a3*a6-a6*a7 == 1 and a1+a2-3*a3+a4+a5+a6+a7==0:
            k=k+1
42     print('
                                {0}'.format(a7)) #ugly code
    have better output
        print('
                                {0}'.format(a6))
        print('The positive real root is {0}, {1}, {2}, {3}, {4}.'.format(
            a1,a2,a3,a4,a5))
        print("")
    print('There are {0} results'.format(k))

```

3. SAGE

3.1. 安装 + 初始代码. Sage 是专为数学家设计的程序, 基于 python, 通过集成大量数据库, 使用其中的数学函数来简化编程难度。我使用的是网上在线的编辑器CoCalc, 在 New 中生成 Sage worksheet. 你可以在每一行前添加sage: 或者不加。在最开始输入需要的宏包, 然后直接进行计算即可。

```
sage: print("Halloworld!233") #The code is the same as
python
```

3.2. 基本逻辑. 本节的基本内容参见[这里](#)。

3.3. 调试. 大部分同 python.

3.4. 参考文档. 官方文档

Sage V9.1 中文文档以及专题文档

Sage Quick Reference Cards

Sage for Undergraduates

基本例子的视频教程

除了 sage 外还有一些数学自动证明软件, 比如Lean theorem prover. 只是我懒不想学自动证明。

3.5. (想保留的) 例子. 这个例子能帮助我计算 Dynkin quiver 的所有不可约表示.

```
sage: Q = DiGraph({1:{2:['a1']},2:{3:['a2']},4:{3:['a3']},5:{4:['a4']},6:{3:['a5']}}).path_semigroup()
sage: M = Q.I(GF(11),3)
sage: M
4 sage: tauM = M.AR_translate()
sage: tauM
```

这个例子能帮助我画出 $\Gamma(5)$ 对应的基本区域.

```
G5 = Gamma(5)
A = FareySymbol(G5).fundamental_domain(show_pairing = true)
show(A,figsize=10,fontsize=10)
```

4. MANIM COMMUNITY

4.1. 安装 + 初始代码. 这是做数学科普视频的软件, 同样基于 python, 其对数学公式的支持远远超过了其他软件, 而且是开源的. 不想安装的懒惰者可以试试在线编译网页 EulerTour.

4.2. 参考文档. 官方文档

中文文档

可以参考hello world 视频教学,常用代码总结/教程

实例:3B1B 的所有动画视频, Dirac's belt trick和自由变形动画。

5. LATEX

5.1. 本文参考文献. listings 的具体设置

fancy 版本的 Mathematica 代码没学会。

L^AT_EX 中如何自动补全代码? 不容易。

本文 python 代码格式来源

5.2. 参考文档. 刘海洋的书:《L^AT_EX 入门》

如何在 latex 里面使用注释效果、tikz 整体调整、颜色名称设置总结

如何在 amsart 下调整目录

5.3. 非初等但是值得学习的宏包.

- Tikz: 数学系画图专属宏包, 其子宏包 Tikzcd 也是相当实用³, 为了简化制图流程大佬们甚至做了两个网站tikzcd-editor和quiver: a modern commutative diagram editor, 都非常实用。另外还可以尝试TikZiT来辅助画图。
- hobby: 画曲线的宏包 (妈妈再也不用担心我画 Tikz 中的曲线)
- tcolorbox: 把文档变得花里胡哨的宏包 (你甚至可以做海报)

5.4. (想保留的) 例子. 这个例子可以画文本宽度的水平直线, 并把公式拉长到文本宽度:

```

\hrule % show text width
2 \resizebox{\textwidth}{!}{ %show the formula very big
    $\displaystyle \frac{1}{\pi}=\frac{2\sqrt{2}}{99^2}
    \sum_{k=0}^{\infty} \frac{(4k)!}{k!^4}\frac{
    {26390k+1103}{396^{4k}}$
}
\begin{gather*} % or small
3.1415926535897932384626433832795028841971693993751
7 058209749445923078164062862089986280348253421170679
821480865132823066470938446095505822317253594081284
8111745028410270193852110555964462294895493038196
\end{gather*}

```

6. MATHEMATICA

6.1. 安装 + 初始代码. Mathematica 是收费的数学计算软件, 参考文档量多但是不够有结构性, 导致我的代码往往是临时性的, 每次计算都需要重新学习代码。(而且我也没有结构性地保存它们) 不过现在可以在这份文档中储存代码了。

6.2. (想保留的) 例子. 这个例子能帮助我计算 affine quiver 相关的矩阵.

```

Cp = ({
    {1, 1, 1, 0, 0, 0},
    {0, 1, 1, 0, 0, 0},
    {0, 0, 1, 0, 0, 0},
5  {0, 0, 1, 1, 0, 0},
    {0, 0, 1, 1, 1, 0},
    {0, 0, 1, 0, 0, 1}
});

```

³tikzcd is in fact only an improvement of matrix in TikZ. Cited from here.

```

Ci = Transpose[Cp];
10 Phi = -Ci.Inverse[Cp]
MatrixPower[Phi, 6]
A = Transpose[Inverse[Ci]] + Inverse[Ci] (*symmetric form*)

```

这个例子计算四元数 \mathbb{H} 上的 non-reduced norm.

```

3 Det[({
    {x, -y, -z, -w},
    {y, x, -w, z},
    {z, w, x, -y},
    {w, -z, y, x}
  )}]
Factor[w^4 + 2 w^2 x^2 + x^4 + 2 w^2 y^2 + 2 x^2 y^2 + y^4
+
8 2 w^2 z^2 + 2 x^2 z^2 + 2 y^2 z^2 + z^4]

```

这个例子画彩色的点.(255 位 RGB)

```

2 ListPlot[{Style[{{1, 1}, {2, 3}, {2, 4}},
    Interpreter["Color"]["RGB 255 0 0"]],
    Style[{{1.5, 1}, {2.3, 3}, {2.4, 4}},
    Interpreter["Color"]["RGB 0 255 0"]}]}

```

7. PUZZLESCRIPT

游戏制作软件:Unity 可以由免费开源游戏引擎Godot替代。简单的 2D 游戏可以先尝试用 PuzzleScript 实现。

请注意, PuzzleScript 只是一个类推箱类游戏制作语言, 比较难制作 3D/扭曲/用鼠标处理的游戏, 而且基本逻辑不完备导致许多功能的实现需要大量技巧。如果不是技术开发狂热者请只把它当成游戏开发初始阶段的调试器使用。

7.1. 基本逻辑. 本节目前主要讨论 RULES. 在 OBJECT 中可以用 transparent 来表示变量 (在正常对象之后), 在 LEGEND 中请将一个字符放在其他变量名之后。

RULES 的基本结构是 keyword1 [keyword2 objectA] -> [objectB] result, 其中

- keyword1 可以取 up/down/left/right, horizontal/vertical, random, late, orthogonal;
- keyword2 可以取 >/</~/v/up/down/left/right, moving/stationary, no, action;

- result可以取 Message, sfx0, win, checkpoint, restart, cancel, again.

否定语句比较困难, 目前只知道部分否定, 如[Background no Player]->[wall]。⁴ 循环语句以 startloop 开始, 以 endloop 结束, 当循环的过程不再改变游戏内容时停止。(这种循环语句是不是智障?)

7.2. 调试. 单行多行注释使用括号(). 快捷键可以参考[这里](#)。

Debug 可以在 Prelude 中写上verbose_logging, 或者功能弱一点的debug. 另外游戏中可以用音效来辅助调试.

7.3. 参考文档. 官方文档 和官方例子

一些例子。差不多类推箱游戏都有 PuzzleScript 的版本。

7.4. (想保留的) 例子. PuzzleScript 网上的例子已经非常方便了。这里只是把网上的例子改着玩。

```

title Basic Objects
author Xiaoxiang Zhou
homepage http://home.ustc.edu.cn/~xx352229/
background_color #222323
5 text_color #f0f6f0
  (color_palette 3)
  debug
  zoomscreen 10x10
  =====
10 OBJECTS
  =====

  Background
  green lightgreen darkgreen
15 00000
  10000
  00010
  00000
  01000
20
  Target
  DarkBlue
  .....
  .000.
25 .0.0.
  .000.

```

⁴通过部分否定以及一些额外的变量即可达到全局的否定, 就是代码丑陋一点罢了。

```
.....  
30 Wall  
   BROWN DARKBROWN  
   00010  
   11111  
   01000  
35 11111  
   00010  
  
   Player  
   darkblue red BLACK  
40 .000.  
   .000.  
   11111  
   .111.  
   .2.2.  
45  
   Crate  
   orange  
   00000  
   0...0  
50 0...0  
   0...0  
   00000  
  
   Spike  
55 black gray white  
   .....  
   ..0..  
   .010.  
   02110  
60 02210  
  
   Stone  
   black white  
   .000.  
65 00110  
   00010  
   00000  
   .000.  
  
70 Angle  
   BROWN red
```

```
00000
0111.
011..
75 01...
0....

0
black
80 .000.
.0.0.
.0.0.
.0.0.
.000.

85
1
black
..0..
.00..
90 ..0..
..0..
.000.

2
95 black
.000.
...0.
.000.
.0...
100 .000.

3
black
.000.
105 ...0.
.000.
...0.
.000.

110 4
black
.0.0.
.0.0.
.000.
115 ...0.
...0.
```

```
5
black
120 .000.
    .0...
    .000.
    ...0.
    .000.
125
6
black
    .000.
    .0...
130 .000.
    .0.0.
    .000.

7
135 black
    .000.
    ...0.
    ...0.
    ...0.
140 ...0.

8
black
    .000.
145 .0.0.
    .000.
    .0.0.
    .000.

150 9
black
    .000.
    .0.0.
    .000.
155 ...0.
    .000.
=====
LEGEND
=====
160 . = Background
```

```

# = Wall
P = Player
* = Crate
165 @ = Crate and Target
O = Target
! = Spike
S = Stone
A = Angle

170 =====
SOUNDS
=====

player Move 33184907
175 Crate Move 412312
EndLevel 123413
Player CantMove 4123412
=====
COLLISIONLAYERS
180 =====

Background
Target
Player, Wall, Crate
185 Spike
0,1,2,3,4,5,6,7,8,9
Stone, Angle

=====
190 RULES
=====

[ > Player | Crate ] -> [ > Player | > Crate ]
late [ Player Spike ] -> [ Spike ]
195 =====
WINCONDITIONS
=====

All Target on Crate
200 =====
LEVELS
=====

205 MESSAGE I make no claim for the original.

```

```

MESSAGE Just some test.

.#p*0@!SA..
210 .#####.
.#..p*.o...
.#####.
01234567890

215 MESSAGE Er. I copied something to check.

#####
#.*00..#
220 #P*...*.#
#..00*.*#
#####

MESSAGE I copied another difficult level. Don't worry.
225
.....####.
#####.p#.
#.....*...#.
#...*##.*#.
230 ###*ooo#.#.
.#.*ooo..#.
.#.#o.o#.#.
.#...#.*.*#
.#*...*....#
235 .#..#####
.####.....

MESSAGE Thank you for playing the game.

```

想要做自己的游戏，先得学会复刻已有的小游戏。做了点尝试。

小技巧：制造边缘砖块的方式（堆叠）以及关卡编辑器中可以使用特殊符号来形象表达，比如LUMirror .

```

title Demake of ORBOX B
author Xiaoxiang Zhou
homepage http://home.ustc.edu.cn/~xx352229/
background_color #222323
5 text_color #f0f6f0
(color_palette 3)
verbose_logging (for better debug)

```



```
youtube CKAc3nYEatw
(realtime_interval 0.15 (automove)
10 again_interval 1 (The amount of time it takes an 'again'
    event to trigger. We won't use it.)
key_repeat_interval 0.10 (hold one key))
=====
OBJECTS
=====
15
Background
green lightgreen  darkgreen
00000
10000
20 00010
00000
01000

Target
25 DarkBlue
.....
.000.
.0.0.
.000.
30 .....

BombObstacle
gray
00000
35 00000
00000
00000
00000

40
Wall
BROWN DARKBROWN
11111
01000
45 11111
00010
11111

WallL
50 Black
0....
```

```

55  0....
    0....
    0....
    0....

    WallR
    Black
    ....0
60  ....0
    ....0
    ....0
    ....0

65  WallU
    Black
    00000
    .....
    .....
70  .....
    .....

    WallD
    Black
75  .....
    .....
    .....
    .....
    00000

80  Wall2
    red
    0...0
    .0.0.
85  ..0..
    .0.0.
    0...0

    Player
90  darkblue red BLACK
    .000.
    .000.
    11111
    .111.
95  .2.2.

```

```
Crate
orange
00000
100 0...0
    0...0
    0...0
    00000

105 LUMirror
    BROWN LightBlue
    ....0
    ...10
110 ..110
    .1110
    00000

    URMirror
115 BROWN LightBlue
    0....
    01...
    011..
    0111.
120 00000

    RDMirror
    BROWN LightBlue
    00000
125 0111.
    011..
    01...
    0....

130 DLMirror
    BROWN LightBlue
    00000
    .1110
    ..110
135 ...10
    ....0

    Bomb
    blue white
140 00000
    01110
```

```
01110
01110
00000
145 Teleport
    blue purple black
    00000
    01110
150 01210
    01110
    00000

    Initial
155 transparent

    Shiftbomb
    transparent

160 =====
    LEGEND
    =====

165 WallOutline = WallU or WallD or WallL or WallR
    Obstacle = Wall or Wall2 or BombObstacle or Bomb
    Mirror    = LUMirror or URMirror or RDMirror or DLMirror
    Solid = Obstacle or Mirror

170 . = Background
    # = Wall
    P = Player and Initial
    * = Crate
    O = Target
175 g = BombObstacle and Target
    d = Wall2
    B = Bomb
    t = Teleport

180 =====
    SOUNDS
    =====
    Sfx0 73682502 (die)
    sfx1 74413905 (mirror)
185 sfx2 89661502 (explosion)
    sfx3 63441708 (walk)
```

```

sfx4 29985308 (teleport)
Sfx10 77515308 (test)
EndLevel 123413
190 =====
COLLISIONLAYERS
=====

Background
195 Mirror, Target, Teleport
Player, Crate, Obstacle
(for beauty)
WallL
WallR
200 WallU
WallD
Initial
Shiftbomb

205 =====
RULES
=====
(Initial setting)
(Editing convenience: Automatically add outlines to walls)
210 [Initial][ WallOutline ] -> [Initial][ ]
up [Initial][ Wall | No Wall ] -> [Initial][ Wall WallU |
]
down [Initial][ Wall | No Wall ] -> [Initial][ Wall WallD |
]
left [Initial][ Wall | No Wall ] -> [Initial][ Wall WallL |
]
right [Initial][ Wall | No Wall ] -> [Initial][ Wall WallR |
]
215 [Initial] ->[]

(begin every step)
(Test if player touches the Bomb)
[ > Player | Bomb ] -> [ Player| ] sfx2
220

(Go directly, for a loop)
(Here stationary is important. Without it the LUMirror can
also move. And the Mirrors must be near the wall.
Otherwise the player would go across the mirror. And be
careful that never make a loop with mirrors and
teleports!)

```

```

right [ > Player | LUMirror] -> [ |up Player stationary
    LUMirror ] sfx1
225 +down [ > Player | LUMirror] -> [ |left Player stationary
    LUMirror ] sfx1
+left [ > Player | URMirror] -> [ |up Player stationary
    URMirror ] sfx1
+down [ > Player | URMirror] -> [ |right Player stationary
    URMirror ] sfx1
+left [ > Player | RDMirror] -> [ |down Player stationary
    RDMirror ] sfx1
+up [ > Player | RDMirror] -> [ |right Player stationary
    RDMirror ] sfx1
230 +right [ > Player | DLMirror] -> [ |down Player stationary
    DLMirror ] sfx1
+up [ > Player | DLMirror] -> [ |left Player stationary
    DLMirror ] sfx1
(now move on)
+ [ > Player| no Obstacle no Mirror] -> [ |> Player ] (
    better if one can wait for some time)
(Teleport)
235 + [> Player Teleport][Teleport no Player] ->[Teleport][
    Teleport > Player] sfx4
(stop)
+ [ > Player| Obstacle] -> [ Player|Obstacle] sfx3
(win condition)
+ [Player Target] -> [stationary Player Target]win
240

random [Bomb][BombObstacle] -> [Bomb][BombObstacle
    Shiftbomb]
[BombObstacle no Shiftbomb] -> []
[Shiftbomb] -> []
245 late [ Player | Wall2 ] -> restart sfx0
(Here I want to wait several seconds to restart)
=====
WINCONDITIONS
=====
250
=====
LEVELS
=====
255 MESSAGE I make no claim for the original.

MESSAGE Level 1

```

```

260 d.....d
    d....#.....d
    d.....#...d
    d..#.....d
    d.....#.d
265 d....p#....d
    d.0.....d
    d.#.....d
    d.....#...d
    d...#.....d
270 d.....d
    d.....d
    d.....d

MESSAGE Level 2
275 d.....d
    d.....#.....d
    d.....#...d
    d...##.....d
    d.#..#.....o.d
280 d.....p#....d
    d.....#...#...d
    d.....#...d
    d..#.....#...d
    d.....#...d
285 d.....d
    d.....d
    d.....d

MESSAGE Level 3
290 d.....d
    d.....###.....d
    d.....#...#...d
    d.....#...d
    d..####...#b...d
295 d.p.bbb..bb#b.o#.d
    d..####...#b...d
    d.....#...d
    d.....#...#...d
    d.....###.....d
300 d.....d
    d.....d
    d.....d
```

```

MESSAGE Level 4
dddddddddddddddddd
d.....d
d..#.....d
d.....#.....d
d.....#.....d
d..#.....d
d..#.....p.....d
d.....#.....d
d...#.....d
d.....#.#.....d
d.....#.....d
d...#.....d
d...o...#.....d
d.....#.#.....d
d.....#.....d
d...#.....d
d.....#.....d
d.....#.....d
dddddddddddddddddd

MESSAGE Level 5
dddddddddddddddddd
d.....d
d.....###.d
d...#.....##.##.d
d.bb....b...#..#.d
d.bb..p..b....g...d
d.....#..#.d
d.....##.##.d
d...#...#.....d
d...#####.....d
d.....d
dddddddddddddddddd

MESSAGE Level 6
dddddddddddddd
d.....d
d.....#.....d
d.....#.....d
d..#.....d
d....bbb.#o...d
d....bpb.#...d

```



```

d....bbb...#.d
d.#.....d
350 d....#.#....d
d....#....d
d....#.....d
d.....d
ddddd
355 MESSAGE Level 7
ddddd
d.....d
d.#####.d
360 d.###...#....d
d.#....#....d
d.#....#....#...d
d.##....#....o....d
d.###.##.... ##### .d
365 d..... ###...# .d
d.##..... #p.#....#d
d.#..t...###...t...#d
d.....#.#...#....#d
d.#.....#.#....###.d
370 d.##....##.... ##### .d
d.....d
ddddd

MESSAGE That's just a test, not a game!
375 ... ##... #
...g ## .t.
...####...
##.####.. #
.# . ## ###
380 .###... ####
.# .bbb#####
.#.bbbbbb...
.# .bbb.....
.#####.
385 .## ...p.t..
.## .....

390

```

```
MESSAGE Thank you for playing the game.
```

8. POV-RAY

8.1. **安装 + 初始代码.** POV-Ray 是一款依赖代码的 3D 建模软件，免费开源，目前个人感觉是“3D 版本的 Tikz”” 一款能够在短时间内持续追踪光线视觉，产生高质量计算机图像的工具。” from here.

程序对大小写敏感。

一些特殊符号：

```
1 o=<0,0,0>, x=<1,0,0>, y=<0,1,0>, z=<0,0,1>, u=<1,0>, v
   =<0,1>.
   sqrt(a), pi, pow(a,n) sin(radians(90)), degrees(atan2(1,1))
```

8.2. **基本逻辑.** 基本几何体请参考这里。

透明背景设置

superellipsoid 可以画圆角圆柱和方体。

8.3. **调试.** 单行注释使用两斜杠//, 多行注释使用/*注释*/。对于.ini 文档，单行注释使用分号;。

关于输出请参考此文档。

8.4. **参考文档.** 官方文档、视频教程、本地 demo、动画效果

POV-Ray 的经典实例是维度：数学漫步。另外也可以参考Mathematical Imagery.

关于 3D 建模还有很多优秀的软件，比如：

- solidwork: 曾经尝试过，后来删了
- 3DMax: 实例可以参考GM 的建模基地。但是收费。
- blender: 免费开源，似乎是 3DMax 的高级替代品，依赖快捷键。

如果我有一台优质的电脑、时间 (没书读时) 和精力那我也会尝试下。现在就算了。

8.5. **(想保留的) 例子.** 这个例子实现了杯子的旋转动画，并制作了一个好的坐标轴。

```
5 // POV-Ray Scene File "begin.pov"
   // by Friedrich A. Lohmueller, Jan-2013, now Xiaoxiang Zhou
   //-----
   global_settings{assumed_gamma 1.0}
   //##default{ finish{ ambient 0.1 diffuse 0.9 }}
   //-----
```

```
10 #include "colors.inc"
#include "textures.inc"
#include "woods.inc"
//-----
// camera -----
camera{
  location <25,25,25>    //best: <25,25,25> <10,-25,25>
15  right -x*image_width/image_height
  sky    <0,0,1>    //see the sky
  look_at <0,0,0>
  angle 22
}
20 // sun -----
light_source{
  <25,25,25>,
  White
  shadowless
25 }
// sky -----
sky_sphere{
  pigment{
    gradient z-3
30    color_map {
      [0 color White]
      [0.95 color Blue]
      [1 color White]
    }
  }
35
  scale 2
  translate <0,0,1>
}
}
40 // axis arrows -----
cone {
  6*x, .1, 6.5*x, 0
}
cone {
45  6*y, .1, 6.5*y, 0
}
cone {
  6*z, .1, 6.5*z, 0
}
50
// axis -----
```

```

cylinder {
    -6*x, 6*x, .021
    pigment{ Black}
55    finish {ambient .3}
}
cylinder {
    -6*y, 6*y, .021
    pigment{ Black}
60    finish {ambient .3}
}
cylinder {
    -6*z, 6*z, .021
    pigment{ Black}
65    finish {ambient .3}
}
    // grid -----
#for (i,-5,5,1)
cylinder {
70    <i,-5,0>, <i,5,0>, .02
    pigment{ Grey}
    finish {ambient .3}
}
cylinder {
75    <-5,i,0>, <5,i,0>, .02
    pigment{ Grey}
    finish {ambient .3}
}
#end
80

text {
    ttf "timrom.ttf" "x" 0.05, 0
    pigment { Black }
85    translate -0.25*x
    rotate<90,0,180>
    translate 6*x-0.8*z
}
text {
90    ttf "timrom.ttf" "y" 0.05, 0
    pigment { Black }
    translate -0.25*x
    rotate<90,0,90>
    translate 6*y-0.8*z
95    }
text {

```

```

    ttf "timrom.ttf" "z" 0.05, 0
    pigment { Black }
    translate -0.25*x
100    rotate<90,0,135>
        translate 6*z-0.8/sqrt(2)*x+0.8/sqrt(2)*y
    }

    difference{
105 union {
    sphere_sweep {
        b_spline
        7,
        <.7,2.2,0>, .8
110        <1.5,.5,0>,.3
        <3.1,.5,0>,.3
        <3.8,2.4,0>,.3
        <3.1,4.1,0>,.3
        <1.5,4.1,0>,.3
115        <.7,2.6,0>,.8
        texture{
        pigment{ Grey}
        finish { phong 1}
    }
120    finish {ambient .3}
        scale <1,1,2.5>
    } //cup
    cylinder{
        <0,0,0>, <0,4.75,0>,2
125
        scale 1/4.75

        texture{
        pigment{ Grey}
130        finish { phong 1}
    }
        scale 4.75
        finish {ambient .4}
        rotate<0,-60,0>
135    }
    } //end of union

    cylinder {
        <0,.2,0>, <0,4.9,0>, 1.8
140        texture{
        pigment{ Grey}

```

```

    finish { phong 1}
  }
    finish {ambient .3}
145 }
rotate<90,0,360*clock>
} //end of difference

```

我们还需要.ini 文档才能够生成动画。

```

; Persistence Of Vision raytracer version 3.7 example file.
2 Antialias=On

Antialias_Threshold=0.1
Antialias_Depth=2
Input_File_Name=begin.pov
7
Initial_Frame=1
Final_Frame=30
Initial_Clock=0
Final_Clock=1
12
Cyclic_Animation=on
Pause_when_Done=off

```

8.6. **Tasks.** 制造一个基本对象集，以及基本操作。

9. 前端,SUBLIME

9.1. **安装 + 初始代码.** 前端没有啥好说的,我所理解的就是 html+css+javascript. 初学者用记事本就可以了,我目前用的是 sublime, 原因请参考这个教程. 代码技术主要参考的是W3(英文版)、Web 技术教程.

9.2. **非初等但是值得学习的库.**

- Apache ECharts: 一个基于 JavaScript 的开源可视化图表库，不过这里的图表限定范围较窄。

10. 电脑快捷键 (中级)

部分参见Chrome 快捷键。

- ctrl+win+left: 切换桌面
- alt+(shift)+tab: 切换任务栏上的程序
- ctrl+(shift)+tab: 切换标签页

- win+D: 显示桌面
- win+left: 页面占左半边
- win+R: 调出运行窗口

11. 正则表达式

正则表达式可以大幅提升搜索和修改代码的效率。

```
1 .{0,100}(?=\d{0,3};\d{0,3}m\n);\d{0,3});(\d{0,3})m\n
```

11.1. 参考文档. 推荐B 站视频。与之配套的练习。

你可以在regexr中测试。

12. 热键设置 (AUTOHOTKEY)

这是一款开源软件。为了 Hackergame 的脚本，同时我猜会大幅度提高打代码的速度。我想可以实现快速打笔记的功能？

12.1. 调试. 注释使用分号;。

12.2. 参考文档. 官方文档. ;

12.3. (想保留的) 例子. 这个例子模拟鼠标和键盘，使用了条件和循环语句。

```
~j::  
Sleep,300  
loop, 1  
{  
5   Random, rand, 1, 999  
    CoordMode, ToolTip  
    Click, 1743 81 1  
    Sleep, 100  
    Click, 36 64 2  
10   Sleep, 100  
    Send, {rand}.0.0.0  
    Sleep, 100  
    Click, 433 323 2  
    Sleep, 100  
15   Send, 0.0.0.0  
    Sleep, 100  
    Click, 452 321 1  
    Sleep, 100  
    if GetKeyState("F10")  
20   {  
    break  
    }  
}
```

```

        Sleep,1000
    }
25 Return

```

这是第二版本，能够 95% 成功提交。

```

#MaxThreadsPerHotkey 3
~j:: ; 热键(可根据您的喜好改变此热键).
#MaxThreadsPerHotkey 1
if KeepWinZRunning ; 这说明一个潜在的线程正在下面的循环中
    运行.
5 {
    KeepWinZRunning := false ; 向那个线程的循环发出停止的
        信号.
    return ; 结束此线程，这样才可以让下面的线程恢复并得知
        上一行所做的更改.
}
; 否则:
10 KeepWinZRunning := true
    rand := 0
    Loop
    {
        ; 以下四行是您要重复的动作(可根据您的需要修改它们):
15 MyNumber := rand . ".233.233.233"
        Click, 1743 81 1
        Sleep, 70
        Click, 74 74 2
        Sleep, 70
20 Send, %MyNumber%
        Sleep, 70
        Click, 429 346 2
        Sleep, 80
        Send, %MyNumber%
25 Sleep, 90
        Click, 1073 31 1
        Send ^{Click 432 360 1}
        Sleep, 190
        rand += 1
30 ; 但请不要修改下面剩下的内容.
        if not KeepWinZRunning ; 用户再次按下 Win-Z 来向循环发
            出停止的信号.
            break ; 跳出此循环.
    }
    KeepWinZRunning := false ; 复位，为下一次使用热键做准备.
35 return

```


第三个版本可以提交 254 个，还剩 2 个。。。

```
#MaxThreadsPerHotkey 3
^j::
#MaxThreadsPerHotkey 1
if KeepWinZRunning
5 {
    KeepWinZRunning := false
    return
}
KeepWinZRunning := true
10 rand := 0
Loop
{
    MyNumber := rand . ".233.233.233"
    Click, 1751 91 1
    15 Sleep, 120
    Click, 74 74 2
    Sleep, 50
    Send, %MyNumber%
    Sleep, 70
    20 Click, 429 346 2
    Sleep, 80
    Send, %MyNumber%
    Sleep, 50
    Click, 587 747 1
    25 ;Click, 1073 31 1
    Send ^{Click 387 437 1}
    Sleep, 60
    rand += 1
    if not KeepWinZRunning
    30 break
}
KeepWinZRunning := false
return
```

最后成功的版本:

```
#MaxThreadsPerHotkey 3
2 ^j::
#MaxThreadsPerHotkey 1
if KeepWinZRunning
{
    KeepWinZRunning := false
    7 return
}
```

```

KeepWinZRunning := true
rand := 0
Loop
12 {
    MyNumber := rand . ".233.233.233"
        Click, 1751 91 1
        Sleep, 120
        Click, 74 74 2
17 Sleep, 50
        Send, %MyNumber%
        Sleep, 70
        Click, 429 346 2
        Sleep, 80
22 Send, %MyNumber%
        Sleep, 40
        Click, 587 747 1
        ;Click, 1073 31 1
        Send ^{Click 387 437 1}
27 Sleep, 60
        rand += 1
        if not KeepWinZRunning
            break
    }
32 KeepWinZRunning := false
return

```

12.4. **Tasks.** 通过该软件创建一个 L^AT_EX 的脚本，要求：

- 自动翻译 iff, SES, LES, mflid, rep...
- 长段代码如 equ+aligned
- 只在打开 TeXstudio 时生效 (使用 IfWinActive+Window Spy)

通过该软件创建一个脚本，要求：

- 自动打开日常软件
- 将其放入开机启动项

已完成，代码如下：

```

#IfWinActive, ahk_class Qt5QWindowIcon
2 ::SES::short exact sequence

::LES::long exact sequence

::TFAE::the following are equivalent:
7

```

```
::iff::if and only if

::st::such that

12 ::eg::for example

::wrt::with respect to

::geo::geometry

17 ::cont::continuous

::diff::differential

22 ::sm::smooth

::mfld::manifold

::lb::line bundle

27 ::mlb::metrized line bundle

::vb::vector bundle

32 ::cplx::complex

::RS::Riemann surface

::AG::algebraic geometry

37 ::agc::algebraic geometric

::AV::abelian variety

42 ::ell::elliptic curve

::proj::projective ;be careful for projection

::inj::injective

47 ::fct::function

::fctor::functor

52 ::iso::isomorphism
```

```

::irr::irreducible

::ind::indecomposable
57  ::adm::admissible

::rep::representation

62  ::repe::representable

::Archi::Archimedean field

::sign::Best wishes,`n`nXiaoxiang Zhou
67  ::sym::symmetric

::corrg::corresponding ;be careful for correspondence

72  ::corre::correspondence

::sset::simplicial set

::LLC::local langlands correspondence
77  ::align:: ;超长变换,我的Texstudio还是有些问题
    (
    \begin{equation*}
    \begin{aligned}
82    =\;&\& \\\
    )

::draw:: ;Texstudio如何原样打出?
    (
87  \begin{figure}[ht]
        \vspace{0cm}
        \centering
        \includegraphics[width=12cm]{}
        \label{fig:}
92    \caption{}
    \end{figure}
    )

::minipage::
97  (

```

```

\begin{figure}[th]
  \begin{minipage}[t]{.48\textwidth}
    \centering
    \includegraphics[width=\textwidth]{}
102    \label{fig1}
  \end{minipage}
  \begin{minipage}[t]{.48\textwidth}
    \centering
    \includegraphics[width=1\textwidth]{}
107    \label{fig2}
  \end{minipage}
\caption{}
\end{figure}
)

112 ^+c:: ;注释
send ^c
sleep, 200
clipboard=%clipboard%
117 tooltip, %clipboard%
sleep, 500
tooltip,
return

122 ::||:: ;临时符号
(
\norm{\cdot}
)

127 ::CLm::Chambert-Loir measure

::dps::\displaystyle

::comm diag::commutative diagram

132 ::reso::resolution

::pfv::partial flag variety

137 ::qgr::quiver Grassmannian

::dv::dimension vector

#IfWinActive
142 ^j::

```

```
Run, "F:\hide\Snipaste\Snipaste.exe"
Run, "F:\hide\WiFi 共享大师\WiFiMaster.exe"
return
```

13. 其他

这里收集乱七八糟的材料，以后说不定可以单独搞个文件。

13.1. 数学的可视化. 这里以四维空间为例。

可以参考Andrew J. Hanson 的主页，也可以参考Github 中的项目(这里有很多可以了解到的就不引用了)。

脑洞：参考四维截面游戏，我们是否可以做个四维版本的小球进洞？

13.2. 数学与游戏. 脑洞：搞个 Riemann surface 的世界，在介绍这个世界的时候以神的旨意来介绍相关概念

1. 安装Unity
2. 用sublime配置unity
3. 制作横版跳跃游戏，包含：
 - 场景/人物/交互/动作(左右动，跳跃，蹲)/人物动画
4. 制作贪吃蛇游戏，熟悉代码(无边框，不蠢，带格子)
5. 制作扭结+贪吃蛇游戏
6. 制作黎曼面游戏(两点)
- 8 方程
 - 游戏类型：找不同？横版跳跃？迷宫？五子棋(4/3子棋，粘附5/4/3子棋)？围棋(扭羊头游戏)？跳棋？空间爬行？扭结+贪吃蛇？塔防？数独？数回？华容道？ORBOX B?(这玩意可以做成三维版本！4维就有点难了)(mini metro？连连看？泡泡龙？消砖块？2048？推箱子？其他的Nikoli Puzzles？)
 - 简易示意图
 - 可能引入的概念：Laurent展开/同调群、基本群的概念(在数回中，请使得最后画出的曲线表示基本群不平凡但是同调群不平凡)/覆盖，万有覆盖，deck transformation(华容道简化难度：允许deck transformation将区块变换至另一个基本区域中相同的位置；或者，我们允许上下粘接)/曲率/曲率驱动游戏/descent(数独解何时能成为更小的黎曼面的解)
 - 视角：收敛区域视角(遮罩圆形)、星形区域视角、基本区域视角、万有覆盖视角、第一人称视角、二维3D视角(视角需要通过"培训+考试"解锁)
- 13 虚拟对话
 - 宇宙到底是啥？(永远都到达不了的点是否真的存在？-black hole 宇宙是几维的？2+3(颜色)+1(时间) 如何测量时间，or时间的流逝是否是均匀的？)

- 自由意志(我可以自由地说我想说的东西, 而你的输入却受到限制)
- 表达自己想做却做不到的事
- 对数学的热爱

脑洞: 利用扭结来制作游戏, 例如“贪吃蛇”。需要的时候考虑离散化, 也可以考虑离散版本的 Reidemeister moves。游戏制作的难点在于如何判定成功 (好的 Goal 是啥。很难去说明一个扭结和另一个扭结等价) 扭结的相关知识可参考这个视频。Ballance 如果能添加磁场和电场, 考虑带电小球 (不考虑电磁感应) 的话就是高中物理的必玩游戏了。

13.3. 数学相关句子模板. 这里收集“啊我之前不知道咋表达, 这就是我当时卡住的时候可以用的句子!”的外文句子。绝对是原创的。

```

1 notations:
Beginning with..., only ... are considered.
shorthand
By an aesthetically desirable abuse of notation,
6 One word about the notation: for simplicity we often omit
...
For notational reasons we usually extend the ... to (
negative indices) by defining ...
The term ... is a convenient, but temporary, expedient.
To save adjectives, if ... is described as ..., it is
implicitly assumed to be ...
Finally, a word about terminology.
11 However on many occasions it is natural or customary to
stay in ... mode when nonetheless one has ... in mind.
While for the most part this convention seems to work well,
it occasionally leads us to make extraneous ...
hypotheses in order to invoke ...
We try to flag this artifice when it occurs.
The following abuse of notation will prove handy:...
16 refer to:
The main references for the material covered in this
section are ...
A more detailed overview of this chapter may be obtained by
reading the introductions to the various sections.
More detailed treatments appear in ..., to which we will
refer for proofs.
mimic the arguments for ...

```

21 We refrain **from** giving a more detailed introduction here
and instead refer the reader to the table of contents as
 well as to the short introductions of the individual
 sections.
 References **for** ... include the monograph ... **and** the more
 introductory account ...

example **and** generalization:
 prototypical example

26 By the same argument as the special case above, ...
 The arguments **in** ... carry over mutatis mutandis.
 provide a navigable route into the area with a complete **and**
 self-contained account of the case ...

eschew **all** generality
 exploit special features to abbreviate/simplify the
 arguments

31 tinker with the examples

Disclaimer:
 minor errors **and** obscurities, a couple of significant
 lapses
 I am by far **not** a person with serious knowledge/
 understanding of ..., thus **in** the ... I may oversimplify
 /overcomplicate things, be inaccurate, **or** even wrong,
and miss subtelties.

36 I have actually been meaning to correct some typos **for** a
while and so I will hopefully do **all** these corrections
in the **next** few weeks.

technical problem:
 achieve some technical control
 Ceci nous contraint à prouver ... par une voie détournée

41 It should be noted that even **if** one **is** interested only **in**
 ..., the proofs often involve more general ...

ask **for** details:
 I find there are a lot of cool ideas **in** your answers, but I
 would be grateful **if** you could be a bit more precise.

46 digression:
 We now want to describe the ... This can be done **in** a more
 straightforward way, but we prefer to include a short
 digression **in** ... theory as this allows us to mention a
 general fact which **is in** the background of a later
 construction anyhow.

Nevertheless, the reader can skip this digression without
loss of continuity **and continue** with ... instead.

excitement/beauty:

51 The literature on ... **is** vast **and** often technical, but the
underlying ideas are possessing of an undeniable beauty.
The excitement provoked by ... stimulated a period of
intense **and** widespread activity.
convey the breadth **and** excitement of the ideas

connect different areas:

56 unveils intricate links between
reveals connections between
is akin to the phenomenon of ...
is deeply intertwined with
give a bird's eye view on

61 waiting for classification:
equivalently(replace i.e.)
encyclopaedic knowledge of ...
In fact, the arguments used in the final part of this proof
, give the following result.

66 Let us unravel this definition a little.(after complicated
definition)
... is pivotal in this matter./ ... play a pivotal rôle.
The arithmetic/information/structure of ... is encapsulated
in ...
encapsulate a significant amount of information concerning
...
We point out the similarities and the differences whenever
appropriate.

71 There is also some degree of novelty in our treatment of
...
... contiennent les outils requis pour surmonter ces
difficultés.
give a sense of the directions in which the area is going.
there is nothing to be gained from specialization at this
stage
take it as a little rebus

76 whirlwind tour of ...
be enamored with
brackets:
{ }:curly set brackets
():round brackets

81 []:square brackets

```
<>:angle brackets
```

```
beginning words: Goals, Pros and Cons
```

另外关于英语语法，可以参考天文物理类英文科技论文写作的常见问题.

13.4. 德语日常句子. 临时存储这里。

```
1 Ich bin pappsatt.  
Der Fisch war nicht üppig.  
Es war ein großer Reinfall!  
Wir müssen feststellen, inwiefern sich die Situation geä  
ndert hat.
```

REFERENCES

SCHOOL OF MATHEMATICAL SCIENCES, UNIVERSITY OF SCIENCE AND TECHNOLOGY
OF CHINA, HEFEI, 230026, P.R. CHINA,
Email address: xx352229@mail.ustc.edu.cn