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)(Comments)
    - 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). 你可以在网上在线的编辑器CoCalc练习 Python.[1]

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。[2]

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)[3]

```
>>> 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:
```

---

[1]但是使用图形化用户界面时往往需要下载。

[2]你需要知道那些类型是可改变的 (mutable);

[3]请小心使用负数的带余除法。

```
5        return False;
    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.** 运行代码的快捷键为 Shift+Enter. 单行注释使用井号`#`, 多行注释使用`'''注释'''or """注释"""`。快捷键为 Ctrl+/.

在 Cocalc 中输入部分单词可以按 Tab 键补全代码，按住 ctrl 键可以多点输入。

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 python 教程

2.5. **可能会学习的标准模块.**

- Tkinter: GUI 图形界面。可以参考教程.

2.6. **(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 $\mathfrak{gl}_n$ 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=a+young_diagram[i]
       for j in range(n):
           if young_diagram[i]<young_diagram[j]:
9              b=b+young_diagram[i]
           else:
```

```
          b=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
7      return False
   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*
```

```python
    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
        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
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
        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))
```

一个失败的计算，尝试画 zeta 函数虚部为 0 的地平线。(不过 Gamma 函数画出来已经比 Mathematica 漂亮了，精度也高一些)

```python
# zeta function: https://mpmath.org/doc/current/functions/zeta.html
#               https://github.com/empet/Math/blob/master/Riemann-Zeta.
    ipynb
# https://stackoverflow.com/questions/36429875/create-mpf-from-array
# contour demo: https://matplotlib.org/stable/gallery/
    images_contours_and_fields/contour_demo.html
import numpy as np
import matplotlib.cm as cm
import matplotlib.pyplot as plt
from scipy.special import gamma, factorial
# importing "cmath" for complex number operations
import cmath


np_zeta = np.frompyfunc(mp.zeta, 1, 1)
```

```
13
    delta = 0.1
    x = np.arange(-15.0, 15.0, delta)
    y = np.arange(-15.0, 15.0, delta)
    X, Y = np.meshgrid(x, y)
18
    Z= gamma(X + 1j*Y).imag
    # np_zeta(X+1j*Y).imag

    fig, ax = plt.subplots()
23  CS = ax.contour(X, Y, Z)
    # ax.clabel(CS, inline=True, fontsize=10)
    # ax.set_title('Simplest default with labels')
```

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。

在矩阵群中的计算可以模仿这个例子。

```
G=SL(2,ZZ)
A = G([0,-1,1,0])
B = G([1,1,0,1])
4  A*B*A^(-1)
```

在指定代数结构的时候可以同时指定生成元。定义数域须先生成多项式环再使用其中的多项式定义。

```
1  R.<t> = PolynomialRing(QQ) #R= Q[t]
   F.<i>=NumberField(t^2+1) #Define number field
   OF=F.ring_of_integers() #Define ring of integers
   [OF.0,OF.1] #Find the basis of ring of integers
   T.<z>= PolynomialRing(OF)
6  L.<l>=NumberField(z^2+i) #define number field in the same
       way again
   LL.<u> = L.absolute_field() #Change the base field
   Q8.<c>= CyclotomicField(8) #Second way to construct field
```

```
Q8.is_isomorphic(LL)
```

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

Sage for Computational Mathematics

基本例子的视频教程

如何使用 GAP 处理群论

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
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 )
```

这个例子能帮助我验证 Hankel determinant 的性质，其中用到了矩阵的构造和形式幂级数.

```
#Here is a check for Hankel determinants
# construction of Matrices https://doc.sagemath.org/html/en
    /reference/matrices/sage/matrix/constructor.html
K.<x> = QQ[[]];
f = (x^3 + 5*x + 13) / (x^3 + 17*x + 2)
ListA = f.coefficients()
A = matrix(QQ, 4, 4, lambda x, y: ListA[x+y+4])
A
det(A)
```

这个例子为了计算常见平面光滑曲线的 bitangles 和 flexs 的数目。其中用到了多项式求值和表格的输出。

```
#Here is a computation of dual curves.
#ref for table: https://doc.sagemath.org/html/en/reference/
    misc/sage/misc/table.html
R.<d> = QQ[];
K=R.fraction_field()
dd = d * (d-1)
u = vector(K, [dd*(dd-1)-d, 1/2 * (dd-1)*(dd-2) - 1/2 * (d
    -1)*(d-2)])
A = matrix(K, [[2,3],[1,1]])
table(columns=[(x,(A^(-1)*u)[0].subs(d=x),(A^(-1)*u)[1].
    subs(d=x)) for x in [2..9]], header_column=["$d$", "$b$"
    , "$f$"], frame=True)
```

这个例子为了实现这里的作业, 练练手感。

```
#This document solves the problems in https://ctnt-summer.
    math.uconn.edu/wp-content/uploads/sites/1632/2020/06/
    CTNT2020-CompNT-Exercises.pdf

# Exercise 5
# https://ask.sagemath.org/question/41952/int-object-has-no
    -attribute-is_prime/
def relative_prime(a,N,X):
    primenumber=0
    for i in range(a,X,N):
        if is_prime(i):
            primenumber=primenumber+1
    return primenumber
relative_prime(1,5,10000)
# v = [(x, relative_prime(1,5,x)) for x in [6,11,..,10000]]
# show(points(v, rgbcolor=(0.2,0.6, 0.1), pointsize=2))
def quotient(X):
    return N(relative_prime(1,5,X)/X*ln(X))
# w = [(x, quotient(x)) for x in [6,26,..,10000]]
# show(points(w, rgbcolor=(0.2,0.6, 0.1), pointsize=2))


# Exercise 6 & 7
from scipy import integrate
def newpi(X):
    sum=0
    for i in range(2,X):
        sum=sum+N(1/ln(i))
```

```
        return sum
27  def Li(X):
        def g(t):
            return N(1/ln(t))
        return integrate.quad(g, 2, X)[0]
    def compare(X):
32      return N(X/ln(X))
    # newpiplot = [(x, newpi(x)) for x in [5,25,..,1000]]
    # oldpiplot = [(x, prime_pi(x)) for x in [5,25,..,1000]]
    # compareplot = [(x, compare(x)) for x in [5,25,..,1000]]
    # Liplot = [(x, Li(x)) for x in [5,25,..,1000]]
37  # show(points(newpiplot, rgbcolor=(0.2,0.6, 0.1), pointsize
        =2)+points(oldpiplot, rgbcolor=(0.2,0.2, 1), pointsize
        =5)+points(compareplot, rgbcolor=(0.9,0.1, 0.1),
        pointsize=2) +points(Liplot, rgbcolor=(0,0, 0),
        pointsize=2))

    compare1plot = [(x, prime_pi(x)/compare(x)) for x in
        [5,25,..,1000]]
    compare2plot = [(x, prime_pi(x)/newpi(x)) for x in
        [5,25,..,1000]]
    compare3plot = [(x, prime_pi(x)/Li(x)) for x in
        [5,25,..,1000]]
42  show(points(compare1plot, rgbcolor=(1,0, 0), pointsize=5)+
        points(compare2plot, rgbcolor=(0,1,0), pointsize=5)+
        points(compare3plot, rgbcolor=(0,0,1), pointsize=5))

    # Exercise 8
    # https://ask.sagemath.org/question/51114/how-can-i-find-
        number-of-elements-of-n-torsion-points-on-an-elliptic-
        curve-over-finite-field/
    # p = 1233.next_prime()
47  # E = EllipticCurve(GF(p),[3,12])
    p = 100.next_prime()**3
    E = EllipticCurve(GF(p),[1,0])
    E.abelian_group().order().factor()
    E.abelian_group()
52
    # Exercise 9
    Qpadic=Qp(53,prec=20) #By changing the number of q, we know
         that \phi_7(x) factors exactly when q \mod 1,2,4,6 mod
         7.
    R.<t> = PolynomialRing(Qpadic)
    g= t^6+t^5+t^4+t^3+t^2+t+1
57  g.factor()
```

```
# Exercise 10
# https://stackoverflow.com/questions/132988/is-there-a-
    difference-between-and-is
Q32.<c>= CyclotomicField(32)
G = Q32.galois_group()
# Some standard test of the group
# G.number_field() is Q32
# G.order()
# G.is_galois()
# G.is_abelian()
# G.is_cyclic()
# G.transitive_label()
# G.list()
for K in G.normal_subgroups():
    if K.order() == 8: # We can't replace "==" with "is"
        print("we've found")
        newfield=K.fixed_field(polred=True)[0]
        print("The number field is a ",newfield,"with
            Galois group of order",newfield.galois_group().
            order())
for K in G.normal_subgroups():
    if K.order() == 4: # We can't replace "==" with "is"
        newfield=K.fixed_field(polred=True)[0]
        if newfield.galois_group().is_cyclic():
            print("Congrats! The number field is a ",
                newfield)
```

这个例子为了实现这里的作业, 练练手感。其中有实现散点/线状图、多数据的作图。

```
0246576f-8987-4797-b380-43a732218552s
def primedivisors(n):
    input1 = divisors(n)
    result = 0
    for i in range(len(input1)):
        if is_prime(input1[i]):
            result=result+1
    return result
# pp1=point([(n,primedivisors(n)) for n in (1..100)],
    rgbcolor=(1,0,0))
# p1=plot(pp1)
# pp2=point([(n,sigma(n,1)) for n in (1..100)],rgbcolor
    =(0,1,0))
# p2=plot(pp2)
```

```
   # pp3=point([(n,euler_phi(n)) for n in (1..100)],rgbcolor
       =(0,0,1))
   # p3=plot(pp3)
15 # p1
   qq1=plot(lambda x:sum(primedivisors(n) for n in (1..floor(x
       ))),(x,1,100),rgbcolor=(1,0,0))
   qguess1=plot(x*log(log(x))+x/7,(x,1,100),rgbcolor=(1,0,0))
   qq1+qguess1
   # qq2=plot(lambda x:sum(sigma(n,1) for n in (1..floor(x)))
       ,(x,1,100),rgbcolor=(1,0,0))
20 # qguess2=plot(x^2/1.23,(x,1,100),rgbcolor=(1,0,0))
   # qq2+qguess2
   # a3=N(1/(2*zeta(2)))
   # qq3=plot(lambda x:sum(euler_phi(n) for n in (1..floor(x))
       ),(x,1,100),rgbcolor=(1,0,0))
   # qguess3=plot(x^2*a3,(x,1,100),rgbcolor=(1,0,0))
25 # qq3+qguess3
```

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. **基本逻辑.** 这里存些常用常忘的代码。
$\overset{\sim}{\longrightarrow}$
$\circlearrowright$
Beamer 从上往下:\begin{overlayarea}{\linewidth}{\textheight}\end{overlayarea}

5.2. **本文参考文献.** listings 的具体设置
fancy 版本的 Mathematica 代码没学会。
LATEX 中如何自动补全代码？ 不容易。
本文 python 代码格式来源

5.3. **参考文档.** 刘海洋的书：《LaTeX 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

上下左右标

tikzcd 中的列对齐和无视文字宽度对齐,全局伸缩以及解决换行问题

5.4. **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[4]，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)

5.5. **(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
\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
3.14159265358979323846264338327950288419716939937510
58209749445923078164062862089986280348253421170679
82148086513282306647093844609550582231725359408128
48111745028410270193852110555964462294895493038196
\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。

---

[4]tikzcd is in fact only an improvement of matrix in TikZ. Cited from here.

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}
    });
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.

```
Det[({
    {x, -y, -z, -w},
    {y, x, -w, z},
3    {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)

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

这个例子画一些亚纯函数的"Dedekind tessellation". 不知道怎么才能提高精度以及把白色部分给填充了.

```
1  ContourPlot[Im[Zeta[x + I y]], {x, -15, 15}, {y, -15, 15},
 Contours -> {0}, Mesh -> All]
ContourPlot[Im[Gamma[x + I y]] == 0, {x, -15, 15}, {y, -15,
    15},
 Contours -> {0}]
Grid[Table[
6   ContourPlot[Im[Gamma[x + I y]], {x, -15, 15}, {y, -15,
    15},
    PlotPoints -> pp,
    MaxRecursion -> mr], {mr, {0, 2}}, {pp, {5, 15}}]]
```

这个例子算 Feynman diagram 的个数.

```
Table[3 Sum[
   m! Binomial[2 i + m, m] Binomial[2 j + m, m] (2 i)!/(
     2^i i!) (2 j)!/(2^j j!), {m, 1, n}, {i, 0, n - m}, {j,
       n - m - i,
     n - m - i}], {n, 10}]
```

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, `global`, 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]`。[5]

RULES 中的每一句都是循环至不再循环为止，如果不允许循环，在该句之前输入`random`; 如果需要多行语句合在一起循环，则从第二行起每行前添加`+`.

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？)

---

[5]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。

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
text_color #f0f6f0
(color_palette 3)
zoomscreen 10x10
enable_level_select
noaction
(
References
https://www.sokobanonline.com/play
http://www.sneezingtiger.com/sokoban/levels/microbanText.
    html
https://www.sohu.com/a/274843792_627888

)
========
OBJECTS
========

Background
green lightgreen   darkgreen
00000
10000
00010
00000
01000

Target
DarkBlue
.....
.000.
.0.0.
.000.
```

```
.....


Wall
BROWN DARKBROWN
00010
11111
01000
11111
00010

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

Crate
orange
00000
0...0
0...0
0...0
00000

Spike
black gray white
.....
..0..
.010.
02110
02210

Stone
black white
.000.
00110
00010
00000
.000.

Angle
BROWN red
```

```
80  00000
    0111.
    011..
    01...
    0....

85  0
    black
    .000.
    .0.0.
    .0.0.
90  .0.0.
    .000.

    1
    black
95  ..0..
    .00..
    ..0..
    ..0..
100 .000.

    2
    black
    .000.
105 ...0.
    .000.
    .0...
    .000.

110 3
    black
    .000.
    ...0.
    .000.
115 ...0.
    .000.

    4
    black
120 .0.0.
    .0.0.
    .000.
    ...0.
    ...0.
```

```
5
black
.OOO.
.O...
.OOO.
...O.
.OOO.

6
black
.OOO.
.O...
.OOO.
.O.O.
.OOO.

7
black
.OOO.
...O.
...O.
...O.
...O.

8
black
.OOO.
.O.O.
.OOO.
.O.O.
.OOO.

9
black
.OOO.
.O.O.
.OOO.
...O.
.OOO.
=======
LEGEND
=======

. = Background
```

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


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


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


     ======
     RULES
     ======

200
     [ > Player | Crate ] -> [ > Player | > Crate ]
     late [ Player Spike ] -> [ Spike ]
     ==============
     WINCONDITIONS
205  ==============


     All Target on Crate


     =======
210  LEVELS
     =======


     level_select_point
     MESSAGE I make no claim for the original.
```

```
MESSAGE Just some test.

.#p*O@!SA..
.#########.
.#..p*.o...
.#########.
01234567890

level_select_point
MESSAGE Er. I copied something to check.


########
#.*OO..#
#P*..*.#
#..OO*.#
########

level_select_point
message If you don't know how to solve this, you can try
    the SokobanAutomaticSolver:  http://m.wodown.com/soft
    /14821.html#downAddress

......####.
#######.p#.
#.....*..#.
#...*##.*#.
##*#ooo#.#.
.#.*ooo..#.
.#.#o.o#.##
.#...#.#*.#
.#*..*....#
.#..#######
.####......
level_select_point

#########
###...###
##......#
##@#@#@.#
##.#p*.##
##.#.#o##
#......##
#...#..##
```

```
       #########
260

       level_select_point

       ##########
       ######..##
265    #.*..*..##
       #.#..o#.##
       #..o.#op.#
       ##*#.@...#
       ##...#####
270    ##########

       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
    enable_level_select
10  (realtime_interval 0.15 (automove)
    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))
    noaction
    ========
15  OBJECTS
    ========

    Background
    green lightgreen  darkgreen
20  00000
    10000
    00010
    00000
    01000
```

```
Target
DarkBlue
.....
.OOO.
.O.O.
.OOO.
.....

BombObstacle
gray
OOOOO
OOOOO
OOOOO
OOOOO
OOOOO


Wall
BROWN DARKBROWN
11111
01000
11111
00010
11111

WallL
Black
O....
O....
O....
O....
O....

WallR
Black
....O
....O
....O
....O
....O

WallU
Black
OOOOO
```

```
70   .....
     .....
     .....
     .....

75   WallD
     Black
     .....
     .....
     .....
80   .....
     00000

     Wall2
     red
85   0...0
     .0.0.
     ..0..
     .0.0.
     0...0
90
     Player
     darkblue red BLACK
     .000.
     .000.
95   11111
     .111.
     .2.2.

     Crate
100  orange
     00000
     0...0
     0...0
     0...0
105  00000


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

```
URMirror
BROWN LightBlue
0....
01...
011..
0111.
00000

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

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

Bomb
blue white
00000
01110
01110
01110
00000

Teleport
blue purple black
00000
01110
01210
01110
00000

Initial
transparent

Shiftbomb
```

```
160  transparent


     =======
     LEGEND
     =======


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

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


     =======
     SOUNDS
     =======
185  Sfx0 73682502 (die)
     sfx1 74413905 (mirror)
     sfx2 89661502 (explosion)
     sfx3 63441708 (walk)
     sfx4 29985308 (teleport)
190  Sfx10 77515308 (test)
     EndLevel 123413
     ================
     COLLISIONLAYERS
     ================
195
     Background
     Mirror, Target, Teleport
     Player, Crate, Obstacle
     (for beauty)
200  WallL
     WallR
     WallU
     WallD
     Initial
```

```
205  Shiftbomb


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


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



     (Go directly, for a loop)
225  (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
     +down  [ > Player | LUMirror] -> [ |left Player stationary
         LUMirror ] sfx1
     +left  [ > Player | URMirror] -> [ |up Player stationary
         URMirror ] sfx1
     +down  [ > Player | URMirror] -> [ |right Player stationary
          URMirror ] sfx1
230  +left  [ > Player | RDMirror] -> [ |down Player stationary
         RDMirror ] sfx1
     +up  [ > Player | RDMirror] -> [ |right Player stationary
         RDMirror ] sfx1
     +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)
+ [> Player Teleport][Teleport no Player] ->[Teleport][
    Teleport > Player] sfx4
(stop)
+  [ > Player| Obstacle] -> [  Player|Obstacle] sfx3
(win condition)
+  [Player Target] -> [stationary Player Target]win


random [Bomb][BombObstacle] -> [Bomb][BombObstacle
    Shiftbomb]
[BombObstacle no Shiftbomb] -> []
[Shiftbomb] -> []
late [ Player | Wall2 ] -> restart sfx0
(Here I want to wait several seconds to restart)
==============
WINCONDITIONS
==============


=======
LEVELS
=======

MESSAGE I make no claim for the original.

LEVEL_SELECT_POINT Level 1

MESSAGE Level 1

dddddddddddd
d.........d
d....#.....d
d.......#..d
d..#.......d
d........#.d
d....p#....d
d.O........d
d.#........d
d......#...d
d...#......d
d.........d
dddddddddddd
```

```
LEVEL_SELECT_POINT Level 2
MESSAGE Level 2
dddddddddddddddd
d..............d
d.......#......d
d..........#...d
d...##.........d
d.#..#........o.d
d.........p#....d
d......#.....#..d
d.......#......d
d..#........#...d
d.........#.....d
d..............d
dddddddddddddddd

LEVEL_SELECT_POINT Level 3
MESSAGE Level 3
dddddddddddddddddd
d...............d
d.......###......d
d......#...#.....d
d............#...d
d..####....#b....d
d.p.bbb..bb#b.o#.d
d..####....#b....d
d............#...d
d......#...#.....d
d.......###......d
d...............d
dddddddddddddddddd

LEVEL_SELECT_POINT Level 4
MESSAGE Level 4
ddddddddddddddddd
d...............d
d..#............d
d.......#.......d
d..........#....d
d.....#.........d
d.#.............d
d..#......#p.....d
d.......#.......d
d....#..........d
```

```
      d........#.#.....d
      d..............#.d
      d.......#.......d
      d...#...........d
325   d....o...#.......d
      d...........#.#..d
      d.......#.......d
      d...#...........d
      d.........#......d
330   d...............d
      dddddddddddddddddd

      LEVEL_SELECT_POINT Level 5
      MESSAGE Level 5
335   dddddddddddddddddddddd
      d...................d
      d...............###.d
      d....#.......##. ##.d
      d.bb.....b...# .. #.d
340   d.bb..p..b.....g....d
      d............# . #..d
      d...........##.##..d
      d...# ... #.........d
      d...#######........d
345   d..................d
      dddddddddddddddddddddd

      LEVEL_SELECT_POINT Level 6
      MESSAGE Level 6
350   ddddddddddddddd
      d...........d
      d.......#....d
      d.....#......d
      d..#.........d
355   d....bbb.#o..d
      d....bpb.#...d
      d....bbb...#.d
      d.#.........d
      d.....#.#....d
360   d.......#....d
      d....#.......d
      d...........d
      ddddddddddddddd

365   LEVEL_SELECT_POINT Level 7
```

```
MESSAGE Level 7
ddddddddddddddddddddddd
d...................d
d.#######...........d
d.###...#...........d
d.#....#...........d
d.#.............#...d
d.##...#......o.....d
d.###..##... #####..d
d.......... ###...#.d
d.##....... #p.#....#.d
d.#..t...###....t...#.d
d.......#.#...#....#.d
d.#.....#. ####...##.d
d.##...##..... #### .d
d...................d
ddddddddddddddddddddddd

LEVEL_SELECT_POINT What?
MESSAGE  That's just a test, not a game!
 ... ##... #
...g ## .t.
 ...####...
##..####.. #
.# . ## ###
.###... ####
.# .bbb#####
.#.bbbbb...
.# .bbb.....
.#########.
.## ...p.t..
.## .......




MESSAGE Thank you for playing the game.
```

以下是黎曼面版本的 2048.

```
title ramified 2048
author Xiaoxiang Zhou
homepage http://home.ustc.edu.cn/~xx352229/
youtube CKAc3nYEatw
```

```
5   verbose_logging (for better debug)
    noaction
    flickscreen 13x6
    (
    ref:
10  https://dario-zubovic.github.io/PuzzleScript/play.html?p=
        e6e4b858fb96486f5d91f390b640345b
    https://gist.github.com/tinjoonatan/
        ada1f82d7a51bad137083c4133ad4fc2
    )
    ========
    OBJECTS
15  ========

    Background
    lightblue

20  wall
    green red
    1.0.1
    .1.1.
    0.1.0
25  .1.1.
    1.0.1

    1
    white lightgrey
30  00000
    00000
    00100
    00000
    00000
35
    2
    white grey
    00000
    00100
40  00000
    00100
    00000

    3
45  white darkgray
    00000
    01000
```

```
00100
00010
00000

4
white black
00000
01010
00000
01010
00000

5
white darkblue
00000
01010
00100
01010
00000

6
white blue
00000
01010
01010
01010
00000

7
white lightblue
00000
01010
01110
01010
00000

8
white lightred
00000
01110
01010
01110
00000

256
```

```
     white red
     00000
95   01110
     01110
     01110
     00000

100  512
     white darkred
     00000
     01110
     01110
105  01110
     00000

     1024
     white orange
110  00000
     01110
     01110
     01110
     00000
115
     2048
     white yellow
     00000
     01110
120  01110
     01110
     00000

     glow
125  yellow
     00000
     0...0
     0...0
     0...0
130  00000

     glow2
     orange
     00000
135  0...0
     0...0
     0...0
```

```
00000

overabove
blue
00000
.....
.....
.....
.....

reach256
transparent

reach512
transparent

reach1024
transparent

copy1
transparent

copy2
transparent

vert
transparent

didmove
transparent

specialcase
transparent

(To fix the screen)
nouseplayer
green red
1.0.1
.1.1.
0.1.0
.1.1.
1.0.1
=======
LEGEND
=======
```

```
Player = 1 or 2 or 3 or 4 or 5 or 6 or 7 or 8 or 256 or 512
    or 1024 or 2048 or nouseplayer


. = Background
# = wall
? = nouseplayer
c = copy1
d = copy2
o = overabove and specialcase and reach256 and reach512 and
    reach1024
s = specialcase
=======
SOUNDS
=======

sfx0 35354707
sfx1 50561103
sfx2 36214503
sfx3 78336708
endgame 11111111111111111111111111111111
endlevel 63354508


================
COLLISIONLAYERS
================

Background
Player, wall
c,d
vert
glow, glow2
overabove
didmove
specialcase
reach256
reach512
reach1024
======
RULES
======
(indication)
random[up Player]->[up Player vert]
random[down Player]->[down Player vert]
```

```
(If one put the vertical arrow, than switsch copy1 and
    copy2)
random [vert][copy1][copy2] -> [vert][copy2][copy1]


[ > player | no Player no wall] -> [ | > player didmove]


[ > 1 | 1 ] -> [ | 2 didmove  ]
[ > 2 | 2 ] -> [ | 3 didmove  ]
[ > 3 | 3 ] -> [ | 4 didmove  ]
[ > 4 | 4 ] -> [ | 5 didmove  ]
[ > 5 | 5 ] -> [ | 6  didmove ]
[ > 6 | 6 ] -> [ | 7  didmove ]
[ > 7 | 7 ] -> [ | 8 didmove  ]
[ > 8 | 8 ] -> [ | 512 didmove  ]
[ > 512 | 512 ] -> [ | 1024 didmove  ]
[ > 1024 | 1024 ] -> [ | 2048 didmove  ]



(horizontal move --- copy1 to copy2)

[copy1 Player][copy2] -> [copy1 Player][copy2 Player]
+ [copy1 no Player][copy2] -> [copy1 no Player][copy2 no
    Player]
+ right[copy1|]->[|copy1]
+ right[copy2|]->[|copy2]

random down [copy1|] ->[|copy1]
random down [copy2|] ->[|copy2]

[copy1 Player][copy2] -> [copy1 Player][copy2 Player]
+ [copy1 no Player][copy2] -> [copy1 no Player][copy2 no
    Player]
+ left[copy1|]->[|copy1]
+ left[copy2|]->[|copy2]

random down [copy1|] ->[|copy1]
random down [copy2|] ->[|copy2]

[copy1 Player][copy2] -> [copy1 Player][copy2 Player]
+ [copy1 no Player][copy2] -> [copy1 no Player][copy2 no
    Player]
+ right[copy1|]->[|copy1]
```

```
+ right[copy2|]->[|copy2]

random down [copy1|] ->[|copy1]
random down [copy2|] ->[|copy2]

[copy1 Player][copy2] -> [copy1 Player][copy2 Player]
+ [copy1 no Player][copy2] -> [copy1 no Player][copy2 no
    Player]
+ left[copy1|]->[|copy1]
+ left[copy2|]->[|copy2]

(twist)
random right [copy1|||] ->[|||copy1]
random right [copy2|||||||||||] ->[|||||||||||copy2]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]
random right [copy1|] ->[|copy1]
random right [copy2|] ->[|copy2]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]
random up [copy1|] ->[|copy1]
random up [copy2|] ->[|copy2]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]
random left [copy1|] ->[|copy1]
random left [copy2|] ->[|copy2]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]

(second twist)
random right [copy1|||||||] ->[|||||||copy1]
random right [|||||||copy2] ->[copy2|||||||]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]
random right [copy1|] ->[|copy1]
```

```
random right [copy2|] ->[|copy2]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]
random down [copy1|] ->[|copy1]
random down [copy2|] ->[|copy2]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]
random left [copy1|] ->[|copy1]
random left [copy2|] ->[|copy2]
random [copy1 Player][copy2] -> [copy1 Player][copy2 Player
    ]
random [copy1 no Player][copy2] -> [copy1 no Player][copy2
    no Player]


(return)
left[copy1|]->[|copy1]
+ left[copy2|]->[|copy2]
random up [copy1|||] ->[|||copy1]
random up [copy2|||] ->[|||copy2]

(If one put the vertical arrow, than switsch back copy1 and
    copy2)
random [vert][copy1][copy2] -> [vert][copy2][copy1]
(destroy indicator)
[vert]-> []

[glow] -> []
random [ no Player no wall ][Player didmove] -> [glow][
    Player] sfx0
[didmove]->[]
[reach256][256]->[][256]sfx1
[reach512][512]->[][512]sfx2
[reach1024][1024]->[][1024]sfx3

up [glow specialcase||||||]->[glow specialcase|||||glow2]
down [glow specialcase||||||]->[glow specialcase|||||glow2]
left [glow2 specialcase||||||||]->[glow2 specialcase|||||||
    glow]
right [glow2 specialcase||||||||]->[glow2 specialcase|||||||
    glow]
```

```
[glow2]->[]

up [glow no specialcase|||||]->[glow|||||glow]
down [glow no specialcase|||||]->[glow|||||glow]
[glow] -> [glow 1]

==============
WINCONDITIONS
==============
some 2048


=======
LEVELS
=======

Message Welcome to the twisted world!

Message The original version of 2048 is here: https://gist.
    github.com/tinjoonatan/ada1f82d7a51bad137083c4133ad4fc2

Message Here you would have a brand new experience of the
    game 2048. It's simpler since there are more spaces for
    you to move.

Message Now enjoy it!

?###########
#c8..###....#
#....###....#
#..oo###..oo#
#..ss###..ss#
############
#d8..###....#
#....###....#
#..ss###..ss#
#..ss###..ss#
############



Message Congratulations!
Message You can make more games on this special twisted
    world! It's a special Riemann surface with constant zero
     curvature.
```

```
Message If you think that is easy, then try this one!


?############
#c8#.###..#.#
#....###....#
#..oo###..oo#
#.#ss###.#ss#
############
#d8#.###..#.#
#....###....#
#..ss###..ss#
#.#ss###.#ss#
############

Message Congratulations! You finally figure it out!
```

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：

```
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。

```
// 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 }}
//------------------------------------------

#include "colors.inc"
#include "textures.inc"
#include "woods.inc"
//------------------------------------------
// camera ----------------------------------
 camera{
  location <25,25,25>    //best: <25,25,25> <10,-25,25>
  right -x*image_width/image_height
  sky   <0,0,1>    //see the sky
  look_at <0,0,0>
  angle 22
 }
// sun ----------------------------------
light_source{
    <25,25,25>,
    White
    shadowless
}
// sky ----------------------------------
sky_sphere{
 pigment{
    gradient z-3
    color_map {
     [0 color White]
     [0.95 color Blue]
     [1 color White]
    }

    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. 调试. 运行代码的快捷键为Ctrl+B. 注释快捷键为Ctrl+/, Ctrl+Shift+/可以在当前位置插入注释。

输入部分单词可以按 Tab 键补全代码，按住 ctrl 键可以多点输入。

对于 java, 可以使用`alert("test");`

9.3. 非初等但是值得学习的库. 对于 JavaScript, 可以从Openbase找到需要的库.

- 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。
- Three.js: 一个三维图形库。

  可以参考geometry-attributes-position

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。

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。

11.2. (想保留的) 例子. 这是在 Hackergame 中的某道题中使用的代码。

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

这是在 latex 中使用的代码，目的是将_good_替换为\textbm{good}.

```
_([^_]*)_
\textbf{\1}
```

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。最后成功的版本：

```
#MaxThreadsPerHotkey 3
^j::
#MaxThreadsPerHotkey 1
if KeepWinZRunning
{
    KeepWinZRunning := false
    return
}
KeepWinZRunning := true
rand := 0
Loop
{
    MyNumber :=  rand . ".233.233.233"
        Click, 1751 91 1
        Sleep, 120
        Click, 74 74 2
        Sleep, 50
        Send, %MyNumber%
        Sleep, 70
        Click, 429 346 2
        Sleep, 80
        Send, %MyNumber%
        Sleep, 40
        Click, 587 747 1
        ;Click, 1073 31 1
        Send ^{Click 387 437 1}
        Sleep, 60
        rand += 1
    if not KeepWinZRunning
```

```
30          break
}
KeepWinZRunning := false
return
```

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

- 自动翻译 iff, SES, LES, mfld, 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

   ::eg::for example

12 ::wrt::with respect to

   ::wlog::without loss of generality

17 ::ptc::particular

   ::fin::finite

   ::fd::finite dimensional

22 ::intro::introduction

   ::geo::geometry

27 ::cont::continuous
```

```
   ::diff::differential
32 ::sm::smooth
   ::mfld::manifold
   ::nbhd::neighbourhood
37
   ::lb::line bundle
   ::mlb::metrized line bundle
42 ::vb::vector bundle
   ::vs::vector space
   ::cplx::complex
47
   ::RS::Riemann surface
   ::AG::algebraic geometry
52 ::agc::algebraic geometric
   ::AV::abelian variety
   ::AR::Auslander--Reiten
57
   ::ARtr::Auslander--Reiten translation
   ::ARth::Auslander--Reiten theory
62 ::ARs::Auslander--Reiten sequence
   ::ARq::Auslander--Reiten quiver
   ::ext::extension
67
   ::ell::elliptic curve
   ::Weq::Weierstrass equation
72 ::proj::projective ;be careful for projection
```

```
::inj::injective

::emb::embedding

::fct::function

::fctor::functor

::iso::isomorphism

::irr::irreducible

::ind::indecomposable

::adm::admissible

::qr::quiver representation

::rep::representation

::repe::representable

::Archi::Archimedean field

::redgp::reductive group

::gp::group

::subgp::subgroup

::sign::Best wishes,`n`nXiaoxiang Zhou

::sym::symmetric

::crspd::corresponding ;be careful for correspondence

::crspdc::correspondence

::equiv::equivalence

::sset::simplicial set

::LLC::local Langlands correspondence

::align:: ;超长变换,我的Texstudio还是有些问题
```

```
    (
    \begin{equation*}
    \begin{aligned}
122   =\;&   \\
    )

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

    ::minipage::
137 (
    \begin{figure}[th]
      \begin{minipage}[t]{.48\textwidth}
        \centering
        \includegraphics[width=\textwidth]{}
142     \label{fig1}
      \end{minipage}
      \begin{minipage}[t]{.48\textwidth}
        \centering
        \includegraphics[width=1\textwidth]{}
147     \label{fig2}
      \end{minipage}
    \caption{}
    \end{figure}
    )
152
    ^+c::  ;注释
    send ^c
    sleep, 200
    clipboard=%clipboard%
157 tooltip, %clipboard%
    sleep, 500
    tooltip,
    return

162 ::||::  ;临时符号
    (
```

```
     \norm{\cdot}
     )

167  ::CLm::Chambert-Loir measure

     ::dps::\displaystyle

     ::comm diag::commutative diagram

172  ::reso::resolution

     ::pfv::partial flag variety

177  ::qpfv::quiver partial flag variety

     ::qgr::quiver Grassmannian

     ::dv::dimension vector

182  ::msm::minimal sectional mono

     ::dgp::dual group

187  ::ldgp::Langlands dual group

     ::Lpm::L-parameter

     ::spgp::split group

192  ::qsgp::quasi-split group

     ::sdpd::semidirect product

197  #IfWinActive
     ^j::
     Run, "F:\hide\Snipaste\Snipaste.exe"
     Run, "F:\hide\WiFi共享大师\WiFiMaster.exe"
     return
```

13. 笔记软件

在这里记录对笔记软件的常用需求。

- 分页
- 可使用不内嵌的网格纸

- 自带文件标签，可分类
- 可在中间插入间隙
- 作图方便，例如圆弧、涂色、表格、组合和固定
- 链接可以点击

14. 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。

Communicating with Interactive Articles: 我想学这个

14.1. **数学的可视化.** 这里以四维空间为例。

可以参考Building 4D polytopes, Andrew J. Hanson 的主页，也可以参考Github 中的项目(这里面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？4D 的根系能不能相对容易地展示？

14.2. **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)
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
    叠，万有覆叠，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、
    万有覆叠视角、第一人称视角、二维3D视角(视角需要通过"培训
    +考试"解锁)
```

```
   虚拟对话
15 -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。

14.3. **数学论文自检流程.** 这里列出自己常见的语法/格式错误清单，以期在下次能减少错误。

- 标点
  - 人名之间用–
  - footnote 写在标点之后
  - 中文文档：同一全半角
  - 引号的修改
  - 行间公式末尾句号, 图表整句标题末尾句号
- 排版
  - 证明符号前移[6]
  - 开头大写，包括图表整句标题、环境后标题

<div align="center">REFERENCES</div>

School of Mathematical Sciences, University of Science and Technology of China, Hefei, 230026, P.R. China,

Email address: xx352229@mail.ustc.edu.cn

---

[6]可以使用\belowdisplayskip=-12pt 或者 align* 环境中的\qedhere 达成目标