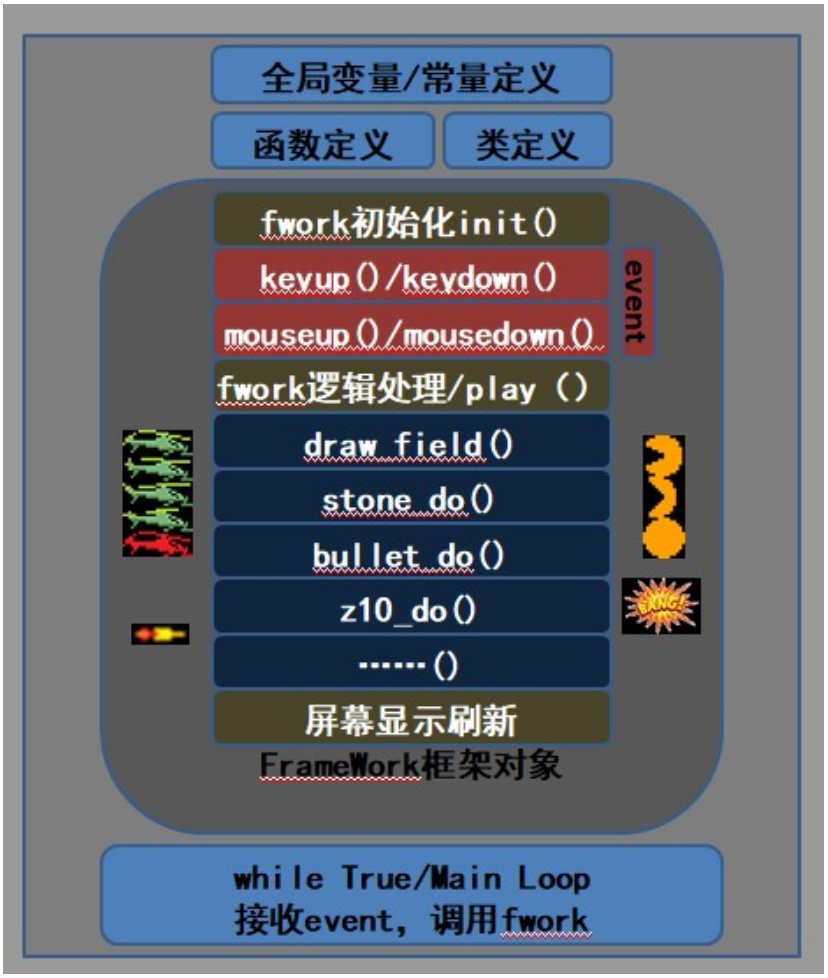
FrameWork 框架



作业：

1、 继续完成“高精度浮点运算”程序；

- 高精度计算：试试是否能让程序进行小数运算
- 提示：把小数转为整数计算，再把结果转为小数
- 比如1.5 与 1.25
  - 1.5 转为：15，1（小数点位置）
  - 1.25转为：125，2（小数点位置）
- 对于加法：
  - 统一小数点位置，1.5，1转为（150，2），与（125，2）统一
  - 调用原有整数加法函数：150+125=175
  - 小数点位置是2，所以输出1.75
- 对于乘法：
  - 调用原有整数乘法函数：15*125=1875
  - 小数点位置应为两个乘数小数点位置之和：1+2=3
  - 所以输出1.875

2、在最新的“直十出击”程序中，尝试增加 enemy。提示：

参考增加 CLS_bullet 和 CLS_bang 的流程。

A）增加 CLS_enemy 类定义，

B）在 CLS_framework 中的 init 增加 enemyList 属性

C）在 CLS_framework 中增加 enemy_do 方法及对此方法的调用；

D） 在 enemy 的产生，在 enemy_do 中随机产生，比如：

if random.random() < 0.01: #0.01 表示每个周期有 1%的可能随机生成一个新 enemy

enemy = CLS_enemy(……)

self.enemyList.append(enemy)

V2.5 代码如下：

```
# V2.5 爆炸效果
import pygame, sys, random
# defination
SCREEN_W, SCREEN_H = 1000, 600
SPACE_UP, SPACE_DOWN = 110, 540 #飞行区域上下边界
SPEEDY_MAX = 5
BG_COLOR , BORDER_COLOR= (0, 0, 80), (80, 80, 80)
G = 0.5 #重力加速度
STONE_H_MIN, STONE_H_MAX, STONE_W = 50, 200, 20 #障碍长短区间与宽度
STONE_SPACE = 160 #障碍间隔
def collide(x1, y1, w1, h1, x2, y2, w2, h2): #add in V2.2
    if x1 + w1 >= x2 and x1 <= x2 + w2 and \
        y1 + h1 >= y2 and y1 <= y2 + h2:
        return True
    else:
        return False
class CLS_gunship( object ): #武装直升机类定义
    def __init__( self, picFile, x, y, w, h, interval, frameNum ):
        pic = pygame.image.load(picFile)  # 图片载入
        pic.set_colorkey( (0,0,0) )         # 设定透明色
        self.pic = pic
        self.x, self.y, self.w, self.h = x, y, w, h # 图片位置与尺寸
        self.interval, self.frameNum = interval, frameNum # 动画速度与帧数
        self.counter = 0 # 动画计数器
        self.speedX = 3
        self.speedY, self.accY = 0, 0
        self.bulletList = [] # add in V2.2
    def move( self ):
        self.speedY += (self.accY + G)
        if self.speedY < -SPEEDY_MAX:
            self.speedY = -SPEEDY_MAX
        elif self.speedY > SPEEDY_MAX:
            self.speedY = SPEEDY_MAX
        self.y += self.speedY
        if self.y < SPACE_UP:
            self.y = SPACE_UP
        elif self.y > SPACE_DOWN - self.h:
            self.y = SPACE_DOWN - self.h
    def draw( self, scr ):
        currentNum = (self.counter // self.interval) % \
                     self.frameNum #当前帧id
        self.counter += 1
        if fwork.status == 1: #add in V2.0
            currentNum = 4
        scr.blit( self.pic, ( self.x, self.y ), \
                  ( 0, currentNum * self.h, self.w, self.h ) )
z10 = CLS_gunship( 'gunship.bmp', 40, 100, 84, 30, 3, 4 ) # Z10对象初始化
```

```
class CLS_stone( object ):     # 障碍类定义
    def __init__( self ):
        self.x, self.w = SCREEN_W, STONE_W
        h = random.randint( STONE_H_MIN, STONE_H_MAX )
        self.h = h
        if h % 2 == 0:       # 利用障碍高度的奇偶性，确定障碍顶天还是立地
            self.y = SPACE_UP
            self.speedY = random.random()*3 #add in V2.4
        else:
            self.y = SPACE_DOWN - h
            self.speedY = -random.random()*3 #add in V2.4
    def move( self ):
        self.x -= z10.speedX #飞机的speedX就是障碍speedX的反向
        #add in V2.4
        self.y += self.speedY
        if self.y < SPACE_UP:
            self.y = SPACE_UP
            self.speedY = -self.speedY
        if self.y > SPACE_DOWN - self.h:
            self.y = SPACE_DOWN - self.h
            self.speedY = -self.speedY
    def draw( self, scr ):
        pygame.draw.rect( scr, ( 80, 80, 80 ), \
                          ( self.x, self.y, self.w, self.h ), 0 )
class CLS_bullet(object): # add in V2.2
    def __init__( self, x, y, speedX, speedY = 0 ): #edit in V2.3
        self.pic = pygame.image.load('bullet.bmp')  # 图片载入
        self.pic.set_colorkey( (0,0,0) )            # 设定透明色
        self.x, self.y = x, y
        self.w, self.h = self.pic.get_size()
        self.speedX = speedX
        self.speedY = speedY #add in V2.3
        self.accX = 0.1
    def move( self ):
        self.speedX += self.accX
        self.x += self.speedX
        #add in V2.3 以下6行copy自gunship.move()
        self.speedY += G/5 #此句稍作修改，bullet无accY
        if self.speedY < -SPEEDY_MAX:
            self.speedY = -SPEEDY_MAX
        elif self.speedY > SPEEDY_MAX:
            self.speedY = SPEEDY_MAX
        self.y += self.speedY
    def draw( self, scr ):
        scr.blit(self.pic, (self.x, self.y))
```

```
class CLS_bang(object): # add in V2.5
    def __init__(self, x, y, speedX, speedY, time): #(x,y)是爆炸中心，time爆炸时长
        self.pic = pygame.image.load('bang.bmp')
        self.x, self.y = x, y
        self.w, self.h = self.pic.get_size()
        self.speedX, self.speedY = speedX, speedY
        self.pic.set_colorkey( (0,0,0) )
        self.time = time #爆炸效果持续周期
        self.timer = 0 #爆炸效果计数器，==time时结束
    def move(self): #爆炸要随着爆炸物移动
        self.x += self.speedX
        self.y += self.speedY
    def draw(self, scr): #带返回值，False表示timer结束
        self.timer += 1
        if self.timer == self.time:
            return False
        rate = self.timer / self.time
        self.img = pygame.transform.scale(self.pic, \
                    (int(rate*self.w/2), int(rate*self.h/2)) )
        w, h = self.img.get_size()
        scr.blit(self.img, (self.x - w / 2, self.y - h / 2))
        return True
class CLS_framework(object): # add in V2.0
    def __init__(self):
        pygame.init()
        self.scr = pygame.display.set_mode( ( SCREEN_W, SCREEN_H ) )
        pygame.display.set_caption('RT GUNSHIP')
        self.clock = pygame.time.Clock()
        self.font = pygame.font.Font(None, 32)
        self.status = 0                         # 0:正常，1:撞击
        self.score = 0
        self.hiscore = 0
        self.stoneList = []                     # 障碍列表
        self.face = pygame.image.load("face.bmp") #add in V2.1
        self.soundBullet = pygame.mixer.Sound("bullet.wav") #add in V2.2
        self.bangList = [] #add in V2.5
        self.soundBang = pygame.mixer.Sound("bang.wav") #add in V2.5
    def play(self):
        if self.status == 1: #crashed状态直接返回
            return
        self.draw_field()
        self.stone_do()
        self.bullet_do() #add in V2.2
        self.bang_do() #add in V2.5
        z10.move()
        z10.draw(self.scr)
        pygame.display.update()                 # 屏幕刷新
        self.clock.tick(100)                    # 帧率可调
```

```
    def draw_field(self):
        self.scr.fill( (0,0,0) )
        pygame.draw.rect(self.scr, BG_COLOR,\
                         (0,SPACE_UP,SCREEN_W, SPACE_DOWN - SPACE_UP),0)
        pygame.draw.rect(self.scr,BORDER_COLOR,(0,SPACE_DOWN, SCREEN_W,10),0)
        pygame.draw.rect(self.scr,BORDER_COLOR,(0,SPACE_UP - 10, SCREEN_W,10),0)
        img = self.font.render('SCORE: ' + str(self.score ), True,\
                               ( 160, 180, 0 ) )
        self.scr.blit( img, ( SCREEN_W - 300, 10 ) )
        # add in V2.1
        img = self.font.render('Hi-SCORE: ' + str(self.hiscore ), True,\
                               ( 0, 180, 240 ) )
        self.scr.blit( img, ( SCREEN_W - 330, 30 ) )
        self.scr.blit( self.face, (0, 0) )
    def stone_do(self):
        lastStoneX = 0
        for stone in self.stoneList:
            stone.move()
            stone.draw( self.scr )
            lastStoneX = stone.x # list里最后一个就是last
            if stone.x + stone.w < 0: # 如果障碍飞出屏幕左侧，则删除该障碍
                self.stoneList.pop( 0 ) #飞出左侧的一定是0号
            if (z10.x + z10.w >= stone.x and \
                z10.x <= stone.x + stone.w): #经过stone
                if (z10.y + z10.h >= stone.y and \
                    z10.y <= stone.y + stone.h):
                    self.status = 1
                else: # 分数逻辑
                    self.score += stone.h*z10.speedX
                    if self.score > self.hiscore:# add in V2.1
                        self.hiscore = self.score
        # 是否需要增加障碍
        if SCREEN_W - lastStoneX \
         > random.randint( STONE_SPACE, STONE_SPACE + \
                           STONE_SPACE // 2 ):
            stone = CLS_stone( )
            self.stoneList.append( stone )
```

```
    def bullet_do(self): #add in V2.2
        lastBulletX = 0
        for b in z10.bulletList:
            lastBulletX = b.x
            b.move()
            b.draw(self.scr)
            for s in self.stoneList:
                if collide(b.x,b.y,b.w,b.h,s.x,s.y,s.w,s.h):
                    s.h -= 20
                    s.y += 20*(s.h%2==1)
                    b.x = SCREEN_W #下次会自动触发删除逻辑
                    self.score -= 100 #击中stone降低游戏难度，所以要扣分
                    bang = CLS_bang(s.x, b.y + 1, \
                                    -z10.speedX, s.speedY, 20)
                    self.bangList.append(bang)
                    self.soundBang.play()
                    break
        if lastBulletX > SCREEN_W:
            z10.bulletList.pop(len(z10.bulletList) - 1)
    def bang_do(self): #add in V2.5
        i = 0
        while i < len(self.bangList):
            bang = self.bangList[i]
            bang.move()
            if bang.draw(self.scr) == False: #爆炸结束，自行了断
                self.bangList.pop(i)
                i -= 1
            i += 1
    def keydown( self, key): #keydown事件处理
        if event.key == pygame.K_UP:
            z10.accY = -1
        if event.key == pygame.K_LEFT:
            z10.speedX -= 1
        if event.key == pygame.K_RIGHT:
            z10.speedX += 1
        if event.key == pygame.K_RETURN: #复活
            self.status , self.score = 0, 0
            self.stoneList = []
        if event.key == pygame.K_SPACE: #飞机shoot
            bullet = CLS_bullet( z10.x + 84, z10.y + 15, 3, -4)
            z10.bulletList.append(bullet)
            bullet = CLS_bullet( z10.x + 84, z10.y + 15, 4, -2)
            z10.bulletList.append(bullet)
            bullet = CLS_bullet( z10.x + 84, z10.y + 15, 5, 0)
            z10.bulletList.append(bullet)
            bullet = CLS_bullet( z10.x + 84, z10.y + 15, 5, -4)
            z10.bulletList.append(bullet)
            bullet = CLS_bullet( z10.x + 84, z10.y + 15, 4 -2)
            z10.bulletList.append(bullet)
            bullet = CLS_bullet( z10.x + 84, z10.y + 15, 3, 0)
            z10.bulletList.append(bullet)
            self.soundBullet.play()
```

```
    def keyup( self, key): #keyup事件处理
        if event.key == pygame.K_UP:
            z10.accY = 0
#-----------------------------------------------------------------
fwork = CLS_framework()  # add in V2.0
#背景音乐 add in V2.1
pygame.mixer.music.load("bg1.mp3")
pygame.mixer.music.set_volume(0.5)
pygame.mixer.music.play(loops=0)
while True: # ----- Main loop -----
    for event in pygame.event.get():  # 事件消息处理
        if event.type == pygame.QUIT: # 关闭窗口事件
            pygame.quit()
            sys.exit()
        if event.type == pygame.KEYDOWN:     # keydown事件处理
            fwork.keydown(event.key)         # edit in V2.0
        elif event.type == pygame.KEYUP:     # keyup键事件处理
            fwork.keyup(event.key)           # edit in V2.0
    fwork.play()  # add in V2.0
```