Pygame实战：传承四十年的封神之作—炸弹人学院游戏【附源码】

原创

顾木子吖 于 2021-09-22 15:18:43 发布 350 收藏 6
分类专栏： Pygame合集 文章标签： pygame python 炸弹人 游戏代码

本文链接：https://blog.csdn.net/weixin_55822277/article/details/120415243
版权

Pygame合集 专栏收录该内容
68 篇文章 65 订阅
订阅专栏

导语

在现在这个浮躁的年代：小编每次登陆王者荣耀，还有每次登陆刺激战场Z！

看着里面的聊天界面，各种代打、各种的找cp。小小编觉得，我们已经失去了玩游戏的初心。

接下来，小台将带领你们走回童年时光，一起领略我们当初玩4399的单纯与天真！

还记得小时候小台每到放学时刻，就会拉着只比我小半岁的小表妹，一块去亲戚家里玩电脑

每一次打开电脑做的第一件事情就是，打开浏览器，输入4399这四个数字，那个时候觉得hao123真是一个神奇的主页！

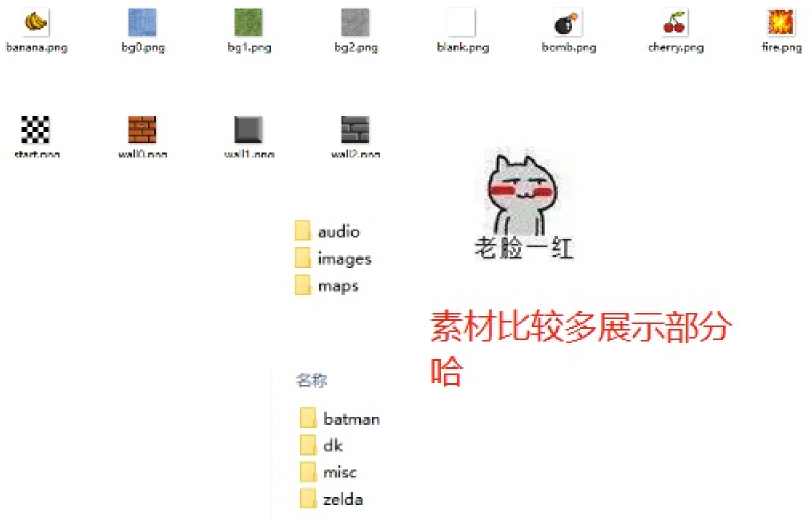
可以让我打开4399玩各种游戏qwq，尤其是Q版泡泡堂深得我心！

那今天就带大家回忆一下童年也做一款经典的炸弹人的小游戏！

正文

游戏规则还清楚哈， 我就不多做介绍了不清楚玩法的可以百度下下！

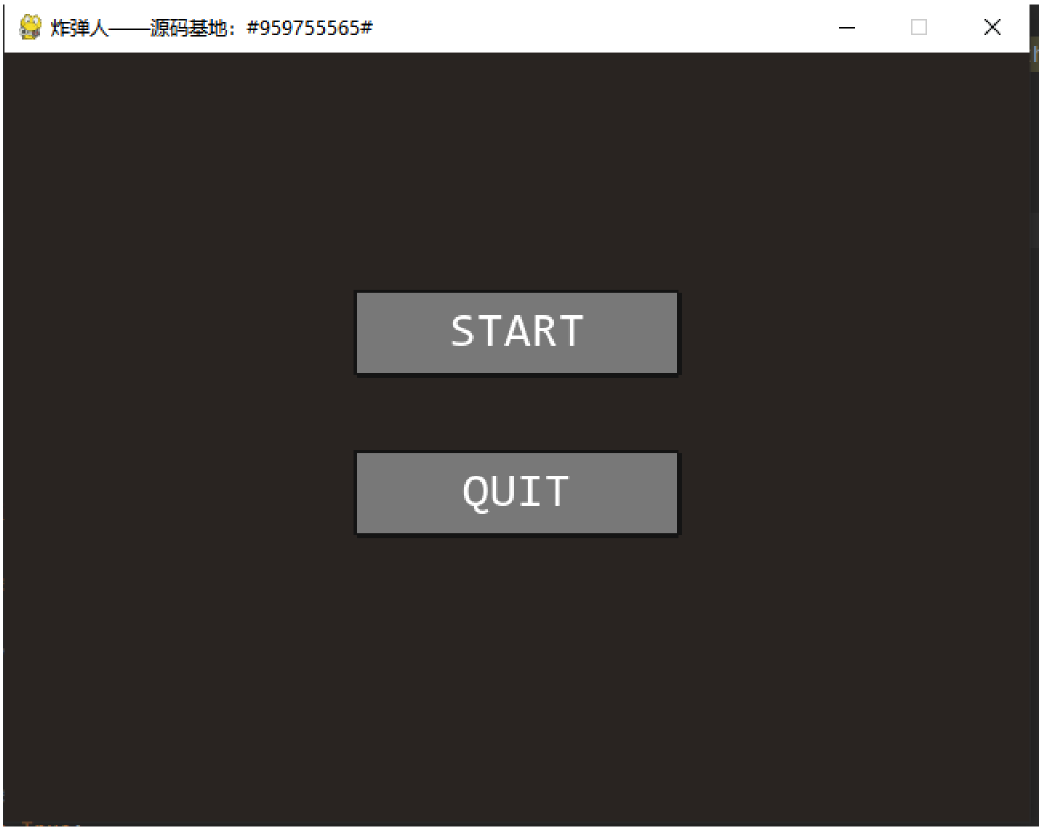
首先准备好相应的素材：【部分如下】



素材比较多展示部分哈

炸弹人主程序：

开始的界面如下：



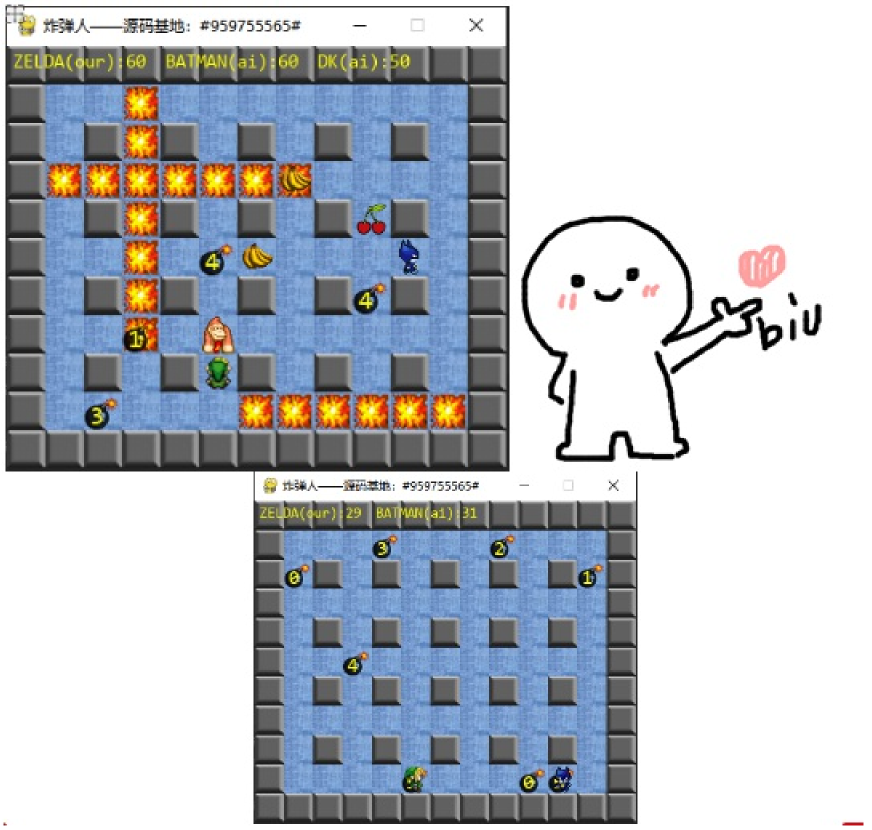
定义地图类：

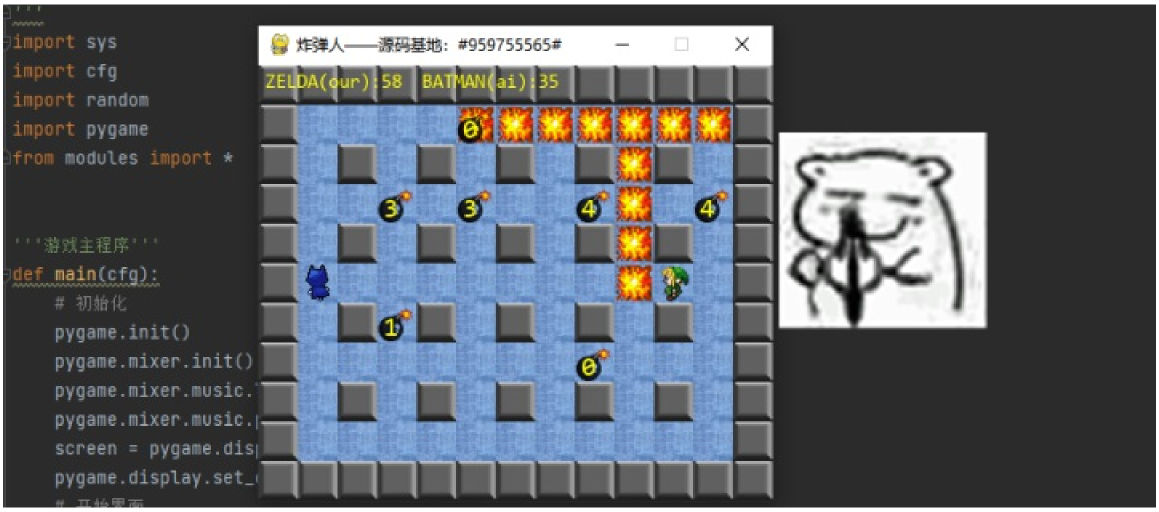
```
class mapParser():
 def __init__(self, mapfilepath, bg_paths, wall_paths, blocksize, **kwargs):
  self.instances_list = self.__parse(mapfilepath)
  self.bg_paths = bg_paths
  self.wall_paths = wall_paths
  self.blocksize = blocksize
  self.height = len(self.instances_list)
  self.width = len(self.instances_list[0])
  self.screen_size = (blocksize * self.width, blocksize * self.height)
 '''地图画到屏幕上'''
 def draw(self, screen):
  for j in range(self.height):
   for i in range(self.width):
    instance = self.instances_list[j][i]
    if instance == 'w':
     elem = Wall(self.wall_paths[0], [i, j], self.blocksize)
    elif instance == 'x':
     elem = Wall(self.wall_paths[1], [i, j], self.blocksize)
    elif instance == 'z':
     elem = Wall(self.wall_paths[2], [i, j], self.blocksize)
    elif instance == '0':
     elem = Background(self.bg_paths[0], [i, j], self.blocksize)
    elif instance == '1':
     elem = Background(self.bg_paths[1], [i, j], self.blocksize)
    elif instance == '2':
     elem = Background(self.bg_paths[2], [i, j], self.blocksize)
    else:
     raise ValueError('instance parse error in mapParser.draw...')
    elem.draw(screen)
 '''随机获取一个空地'''
 def randomGetSpace(self, used_spaces=None):
  while True:
   i = random.randint(0, self.width-1)
   j = random.randint(0, self.height-1)
   coordinate = [i, j]
   if used_spaces and coordinate in used_spaces:
    continue
   instance = self.instances_list[j][i]
   if instance in ['0', '1', '2']:
    break
  return coordinate
 '''根据坐标获取元素类型'''
 def getElemByCoordinate(self, coordinate):
  return self.instances_list[coordinate[1]][coordinate[0]]
 '''解析.map文件'''
 def __parse(self, mapfilepath):
  instances_list = []
  with open(mapfilepath) as f:
   for line in f.readlines():
    instances_line_list = []
    for c in line:
     if c in ['w', 'x', 'z', '0', '1', '2']:
      instances_line_list.append(c)
    instances_list.append(instances_line_list)
  return instances_list
```

定义必要的一些精灵类：角色，水果等等。

```
'''墙类'''
class Wall(pygame.sprite.Sprite):
 def __init__(self, imagepath, coordinate, blocksize, **kwargs):
  pygame.sprite.Sprite.__init__(self)
  self.image = pygame.image.load(imagepath)
  self.image = pygame.transform.scale(self.image, (blocksize, blocksize))
  self.rect = self.image.get_rect()
  self.rect.left, self.rect.top = coordinate[0] * blocksize, coordinate[1] * blocksize
  self.coordinate = coordinate
  self.blocksize = blocksize
 '''画到屏幕上'''
 def draw(self, screen):
  screen.blit(self.image, self.rect)
  return True


'''背景类'''
class Background(pygame.sprite.Sprite):
 def __init__(self, imagepath, coordinate, blocksize, **kwargs):
  pygame.sprite.Sprite.__init__(self)
  self.image = pygame.image.load(imagepath)
  self.image = pygame.transform.scale(self.image, (blocksize, blocksize))
  self.rect = self.image.get_rect()
  self.rect.left, self.rect.top = coordinate[0] * blocksize, coordinate[1] * blocksize
  self.coordinate = coordinate
  self.blocksize = blocksize
 '''画到屏幕上'''
 def draw(self, screen):
  screen.blit(self.image, self.rect)
  return True


'''水果类'''
class Fruit(pygame.sprite.Sprite):
 def __init__(self, imagepath, coordinate, blocksize, **kwargs):
  pygame.sprite.Sprite.__init__(self)
  self.kind = imagepath.split('/')[-1].split('.')[0]
  if self.kind == 'banana':
   self.value = 5
  elif self.kind == 'cherry':
   self.value = 10
  else:
   raise ValueError('Unknow fruit %s...' % self.kind)
  self.image = pygame.image.load(imagepath)
  self.image = pygame.transform.scale(self.image, (blocksize, blocksize))
  self.rect = self.image.get_rect()
  self.rect.left, self.rect.top = coordinate[0] * blocksize, coordinate[1] * blocksize
  self.coordinate = coordinate
  self.blocksize = blocksize
 '''画到屏幕上'''
 def draw(self, screen):
  screen.blit(self.image, self.rect)
  return True


'''炸弹类'''
class Bomb(pygame.sprite.Sprite):
 def __init__(self, imagepath, coordinate, blocksize, digitalcolor, explode_imagepath, **kwargs):
  pygame.sprite.Sprite.__init__(self)
  self.image = pygame.image.load(imagepath)
  self.image = pygame.transform.scale(self.image, (blocksize, blocksize))
  self.explode_imagepath = explode_imagepath
  self.rect = self.image.get_rect()
  # 像素位置
  self.rect.left, self.rect.top = coordinate[0] * blocksize, coordinate[1] * blocksize
  # 坐标(元素块为单位长度)
  self.coordinate = coordinate
  self.blocksize = blocksize
  # 爆炸倒计时
  self.explode_millisecond = 6000 * 1 - 1
  self.explode_second = int(self.explode_millisecond / 1000)
  self.start_explode = False
  # 爆炸持续时间
  self.exploding_count = 1000 * 1
  # 炸弹伤害能力
  self.harm_value = 1
  # 该炸弹是否还存在
  self.is_being = True
  self.font = pygame.font.SysFont('Consolas', 20)
  self.digitalcolor = digitalcolor
 '''画到屏幕上'''
 def draw(self, screen, dt, map_parser):
  if not self.start_explode:
   # 爆炸倒计时
   self.explode_millisecond -= dt
   self.explode_second = int(self.explode_millisecond / 1000)
   if self.explode_millisecond < 0:
    self.start_explode = True
   screen.blit(self.image, self.rect)
   text = self.font.render(str(self.explode_second), True, self.digitalcolor)
   rect = text.get_rect(center=(self.rect.centerx-5, self.rect.centery+5))
   screen.blit(text, rect)
   return False
  else:
   # 爆炸持续倒计时
   self.exploding_count -= dt
   if self.exploding_count > 0:
    return self.__explode(screen, map_parser)
   else:
    self.is_being = False
    return False
 '''爆炸效果'''
 def __explode(self, screen, map_parser):
  explode_area = self.__calcExplodeArea(map_parser.instances_list)
  for each in explode_area:
   image = pygame.image.load(self.explode_imagepath)
   image = pygame.transform.scale(image, (self.blocksize, self.blocksize))
   rect = image.get_rect()
   rect.left, rect.top = each[0] * self.blocksize, each[1] * self.blocksize
   screen.blit(image, rect)
  return explode_area
 '''计算爆炸区域'''
 def __calcExplodeArea(self, instances_list):
  explode_area = []
  # 区域计算规则为墙可以阻止爆炸扩散，且爆炸范围仅在游戏地图范围内
  for ymin in range(self.coordinate[1], self.coordinate[1]-5, -1):
   if ymin < 0 or instances_list[ymin][self.coordinate[0]] in ['w', 'x', 'z']:
    break
   explode_area.append([self.coordinate[0], ymin])
  for ymax in range(self.coordinate[1]+1, self.coordinate[1]+5):
   if ymax >= len(instances_list) or instances_list[ymax][self.coordinate[0]] in ['w', 'x', 'z']:
    break
   explode_area.append([self.coordinate[0], ymax])
  for xmin in range(self.coordinate[0], self.coordinate[0]-5, -1):
   if xmin < 0 or instances_list[self.coordinate[1]][xmin] in ['w', 'x', 'z']:
    break
   explode_area.append([xmin, self.coordinate[1]])
  for xmax in range(self.coordinate[0]+1, self.coordinate[0]+5):
   if xmax >= len(instances_list[0]) or instances_list[self.coordinate[1]][xmax] in ['w', 'x', 'z']:
    break
   explode_area.append([xmax, self.coordinate[1]])
  return explode_area


'''角色类'''
class Hero(pygame.sprite.Sprite):
 def __init__(self, imagepaths, coordinate, blocksize, map_parser, **kwargs):
  pygame.sprite.Sprite.__init__(self)
  self.imagepaths = imagepaths
  self.image = pygame.image.load(imagepaths[-1])
  self.image = pygame.transform.scale(self.image, (blocksize, blocksize))
  self.rect = self.image.get_rect()
  self.rect.left, self.rect.top = coordinate[0] * blocksize, coordinate[1] * blocksize
  self.coordinate = coordinate
  self.blocksize = blocksize
  self.map_parser = map_parser
  self.hero_name = kwargs.get('hero_name')
  # 生命值
  self.health_value = 50
  # 炸弹冷却时间
  self.bomb_cooling_time = 5000
  self.bomb_cooling_count = 0
  # 随机移动冷却时间(仅AI电脑用)
  self.randommove_cooling_time = 100
  self.randommove_cooling_count = 0
 '''角色移动'''
 def move(self, direction):
  self.__updateImage(direction)
  if direction == 'left':
   if self.coordinate[0]-1 < 0 or self.map_parser.getElemByCoordinate([self.coordinate[0]-1, self.coordinat
    return False
   self.coordinate[0] = self.coordinate[0] - 1
  elif direction == 'right':
   if self.coordinate[0]+1 >= self.map_parser.width or self.map_parser.getElemByCoordinate([self.coordinate
    return False
   self.coordinate[0] = self.coordinate[0] + 1
  elif direction == 'up':
   if self.coordinate[1]-1 < 0 or self.map_parser.getElemByCoordinate([self.coordinate[0], self.coordinate[
    return False
   self.coordinate[1] = self.coordinate[1] - 1
  elif direction == 'down':
   if self.coordinate[1]+1 >= self.map_parser.height or self.map_parser.getElemByCoordinate([self.coordinat
    return False
   self.coordinate[1] = self.coordinate[1] + 1
  else:
   raise ValueError('Unknow direction %s...' % direction)
  self.rect.left, self.rect.top = self.coordinate[0] * self.blocksize, self.coordinate[1] * self.blocksize
  return True
 return True
'''随机行动(AI电脑用)'''
def randomAction(self, dt):
 # 冷却倒计时
 if self.randommove_cooling_count > 0:
  self.randommove_cooling_count -= dt
 action = random.choice(['left', 'left', 'right', 'right', 'up', 'up', 'down', 'down', 'dropbomb'])
 flag = False
 if action in ['left', 'right', 'up', 'down']:
  if self.randommove_cooling_count <= 0:
   flag = True
   self.move(action)
   self.randommove_cooling_count = self.randommove_cooling_time
 elif action in ['dropbomb']:
  if self.bomb_cooling_count <= 0:
   flag = True
   self.bomb_cooling_count = self.bomb_cooling_time
 return action, flag
'''生成炸弹'''
def generateBomb(self, imagepath, digitalcolor, explode_imagepath):
 return Bomb(imagepath=imagepath, coordinate=copy.deepcopy(self.coordinate), blocksize=self.blocksize, dig
'''画到屏幕上'''
def draw(self, screen, dt):
 # 冷却倒计时
 if self.bomb_cooling_count > 0:
  self.bomb_cooling_count -= dt
 screen.blit(self.image, self.rect)
 return True
'''吃水果'''
def eatFruit(self, fruit_sprite_group):
 eaten_fruit = pygame.sprite.spritecollide(self, fruit_sprite_group, True, None)
 for fruit in eaten_fruit:
  self.health_value += fruit.value
'''更新角色朝向'''
def __updateImage(self, direction):
 directions = ['left', 'right', 'up', 'down']
 idx = directions.index(direction)
 self.image = pygame.image.load(self.imagepaths[idx])
 self.image = pygame.transform.scale(self.image, (self.blocksize, self.blocksize))
```

效果如下：


炸弹人——源码基地：#959755565#
ZELDA(our):60 BATMAN(ai):60 DK(ai):50
biu
炸弹人——源码基地：#959755565#
ZELDA(our):29 BATMAN(ai):31



```
'''
import sys
import cfg
import random
import pygame
from modules import *


'''游戏主程序'''
def main(cfg):
    # 初始化
    pygame.init()
    pygame.mixer.init()
    pygame.mixer.music.
    pygame.mixer.music.
    screen = pygame.disp
    pygame.display.set_
    # 开始界面
```

这精致的画面还可以吧~哈哈哈啊 快夸我快夸我~

总结

安啦！文章就写到这里，你们的支持是我最大的动力，记得三连哦~

□关注小编——获取更多精彩内容哦！

温馨提示：不要忘记点赞、关注、评论哦！看我手上的炸弹~