逃!

作者：霍禹佳、高铭星、朱子仪、梁鞍華

[摘要] 本作融合了企鹅、史诗英雄故事、解谜和游戏这四种元素，创造出一款全新的解谜类游戏。通过对故事、谜题、画面艺朮、操作等内容的精心设计，本作将满足玩家极大的好奇心。为了迎合本课程发挥想象的宗旨，在代码实现上，本作仅利用 Python 自有的模块以及 pygamezero 和 random 模块，以求以有限的工具实现更多的功能。最后，我们也展望将来，期望完善和扩展游戏内容以及实现不同设备上的移植。

一、选题及创意介绍

本作的选题和创意来源于我们四个所热爱的事物——企鹅、史诗英雄故事、解谜、游戏。我们在本作中试图融合这四种元素，在基于企鹅的个体生物特征与种群特征、具有历史性的故事文本、故事与谜题的关系及其拓展等主题的研究和探索上，我们开发出一个以企鹅为主角，以动物园为主要场景，以企鹅拟人性宗教哲学性思考为核心思想，以故事为导向的解谜类游戏——**《逃!》**。

本作以故事情节为主要游戏主线，在游戏里，玩家将浸淫在一个全新的故事当中。通过代入和操作主角 Skipper，玩家将作为一只企鹅来以一个不同的视角去看待真实世界发生的问题。本作极大地使谜题和剧情融为一体，并且，谜题的设计考验了玩家在记忆力、控制力、逻辑推理各个方面的能力。玩家对谜题的解答也就是对故事发展的选择，不同的选择将决定故事不同的最终走向，只有通过所有的谜题，玩家才能一睹企鹅“世界”的真相，并走出只属于自己的结局。

二、设计方案

我们的设计方案由三部分构成：游戏剧本、谜题、绘图、画面结构。

1. 故事

本作的游戏模型是典型的珍珠串模型，因而在故事文本的选择上比较单一且封闭。本作游戏故事开始于一个动物园里老企鹅所讲述的一个史诗故事，这个古老的故事讲述了企鹅种群尚处于战国时代时的某一个国王走出“牢笼”并把从“外面世界”所获得的宝藏藏在一处“遥远的他方”的一段“历史”。年轻的企鹅 Skipper 有一天幸运的获得了“藏宝图”，并由于展开了冒险旅程。最终，这张 “藏宝图”将把 Skipper 带向世界的真相……

2. 谜题

本作内嵌的小游戏目前有四个，分别是躲激光、对对碰、数独和走迷宫。

2.1 躲激光

设置两个 Actor 作为射线，它们将绕固定的端点作 160°(来回的)旋转。利用 A, D 键控制 Skipper 在画面中左右走动，S 键控制 Skipper 滑行，W 键使企鹅直立，通过空格键触发企鹅的动作交替，并沿着企鹅的朝向移动相应的距离。如果企鹅与射线发生碰撞，游戏失败，从游戏左侧移动到右侧则成功。

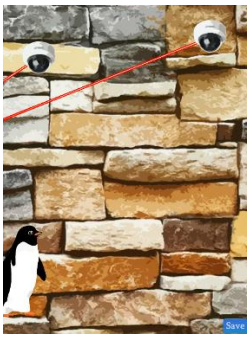


图 1：躲激光

2.2 对对碰

该小游戏由 4×3 的方格组成，包含了 6 对不同的图片，开始时图片全部被遮蔽，玩家用鼠标点击一下图片则显示图片正面，只有完成配对之后两张图片才会消失，否则就会复原，考验了玩家的记忆力。完成所有配对即可继续前进。

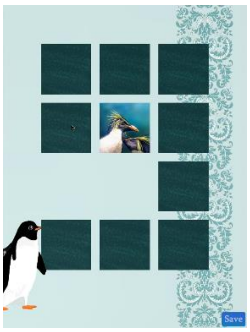


图 2：对对碰

2.3 数独

游戏每次都会生成随机的 9×9 数独，可以通过设置空格数确定难度(默认 50 个空格)，玩家点击方格并输入数字，若所填充的数字与正确数字不同，则所有错误数字显示红色，否则显示代表正确的绿色。当没有空格的时候进入下一关。

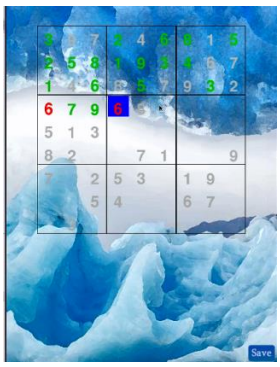


图 3：数独

2.4 迷宫

迷宫由 31×21 个方格组成，Skipper 需要从迷宫的左上角走到左下角。玩家使用上下左右键来控制 Skipper 走动，但 Skipper 只能在蓝色方格上行走，若碰到黑色方格，则原地不动。迷宫设有两条通关路径，若走第一条路径，则游戏失败，走第二条路径才能通关。

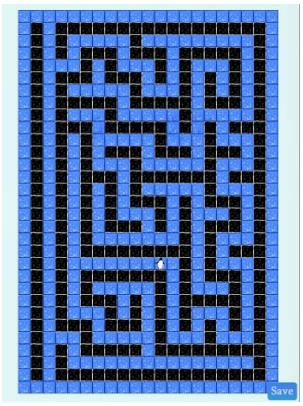


图 4：迷宫

3. 绘图

画风方面，我们选用了较为具有真实感的水彩美术风格，该美术风格的特点在于刻画和光影变化比较注重写实，起稿的时候多用单色或者纯色的色块来找型，笔触也像油画一样一层一层的覆盖逐渐细化，因而可以增强玩家的代入感。

4. 画面结构

画面大小方面，我们由于最终想实现的是一款手机游戏，因此选用竖屏 585:780 的比例，以方便在时机成熟时把本作移植到手机上，目前本作只能够在电脑上运行。

5. 存档读档

本作设有存读档系统，玩家可随时退出游戏而不必重要重新开始游戏。当然，本作也设有重启功能，以方便玩家在游戏过程中途重新开始游戏。

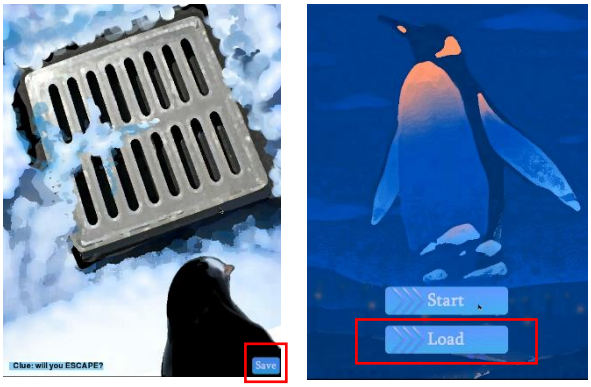


图 5：存档与读档

三、实现方案与代码分析

我们的游戏设置了多个场景，整个游戏的流程图如下：

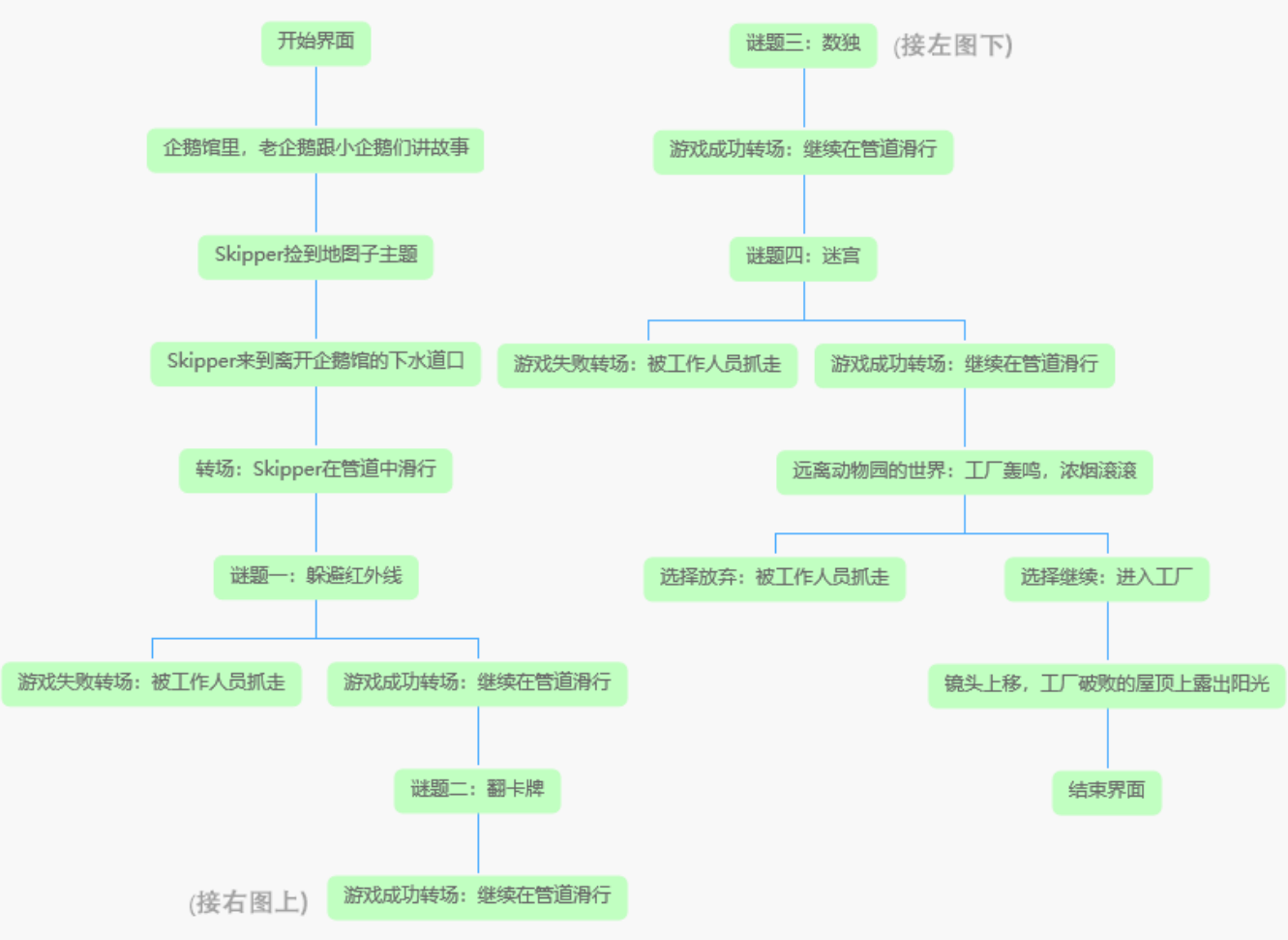


图 6：游戏整体流程图

代码实现上，全局变量就不再赘述。整体结构上，为了让各个场景之间可以转换，我们设置了一个 Actor，取名为 penguin，penguin 下定义了一个变量 scene，这个变量会记录每个场景对应的背景图，scene 改变代表游戏的场景变化。为了处理各个场景，给每个场景都定义了一个 draw 函数(取名为 draw+场景编号)，由游戏总的 draw()调用，使每次场景变化之后按照函数的定义进行场景的渲染。场景的改变可以分为两种方式：一是通过按键触发，一是在某个场景停留的时间达到预设，就会自动切换到下一个场景。

```
def draw():
    screen.clear()
    screen.blit(penguin.scene, (0, 0))
    if penguin.scene == 'start':  draw1()
    elif penguin.scene == 'telling_stories':  draw2()
    elif penguin.scene == 'by_accident2':  draw3()
```

```
    elif penguin.scene == 'escape':  draw4()
    elif penguin.scene == 'puzzle':  drawpuzzle1()
    elif penguin.scene == 'puzzle1':  drawpuzzle2()
    elif penguin.scene == 'puzzle3':  drawpuzzle4()
    elif penguin.scene == 'puzzle2':  drawpuzzle3()
    elif penguin.scene == 'polluted_factory':  draw5()
    elif penguin.scene == 'factory0':
        if not penguin.factoryappear:  draw6()
        else:  draw7()
    if penguin.appear:  penguin.draw()
    if penguin.scene != 'start' and penguin.scene != 'the_end':
save.draw()
```

代码 1：draw()函数

对于 update 函数，我们采取了相似的写法：根据 penguin.scene 指示的场景不同，记录帧数的 penguin.cnt 以及每个场景对应物体的运动也不同，为了使各个部件显示的时候不会过于突兀，我们设计了延时，使部件总是不会立即出现。为节省篇幅，各个场景的具体处理不再展示，用注释代替。

```
def update():
  if penguin.scene == 'telling_stories':
        '''处理：penguin.cnt 记录到 360 时场景跳转'''
    elif penguin.scene == 'shadow':
        '''penguin.cnt 记录到 300 时跳转，上一个 scene 不同跳转的场景不同'''
    elif penguin.scene == 'nakeshadow':
        '''penguin.cnt 记录到 300 时跳转，上一个 scene 不同跳转的场景不同'''
    elif penguin.scene == 'puzzle':
        '''处理：两道射线的旋转(-80°到 80°)'''
        '''当企鹅穿过整个页面没有碰到射线的时候场景跳转'''
    elif penguin.scene == 'polluted_factory':
        '''处理：penguin.cnt 记录到 300 时显示不同的选择和游戏提示'''
    elif penguin.scene == 'factory0':
        '''处理：penguin.cnt 记录到 180 时显示不同的选择和游戏提示'''
    if factory.stop and not (penguin.scene == 'the_end'):
        '''处理：penguin.cnt 记录到 777 时场景跳转'''
    if player.success and penguin.scene == 'puzzle2':
        '''处理：不同的场景不同'''
    if penguin.sudoku and penguin.scene == 'puzzle3':
        '''''专门处理数独成功之后的延时'''
```

代码 2：update()函数结构

主线处理完毕之后，这里展示了我们所有其他的辅助函数和功能模块：

```
on_mouse_down(pos) '''处理各个 Actor 被鼠标点击之后的变化'''
on_key_down(key) '''处理键盘的信号'''

# 以下 7 个函数用来处理企鹅的移动
set_move()
set_slip()
set_slip_normal()
set_walk()
set_walk_normal()
set_backwalk()
set_backwalk_normal()

# 以下 7 个函数用来处理数独游戏
print_matrix(matrix) '''在游戏页面中画出生成的数独'''
shuffle_number(_list) '''打乱数字'''
check(matrix, i, j, number) '''检查填入的数字是否正确'''
build_game(matrix, i, j, number)
give_me_a_game(blank_size=9) '''以上 2 个函数用来生成随机的数独'''
check_win(matrix_all, matrix) '''检查玩家是否完成整个游戏'''
check_color(matrix, i, j) '''确定鼠标选中的方格的颜色'''

# 以下函数用来处理每个场景的展示要求
draw1() '''开始界面'''
draw2() '''老企鹅讲故事'''
draw3() '''遇到地图'''
draw4() '''通过下水道逃跑'''
draw5() '''污染工厂区'''
draw6() '''进入工厂后'''
draw7() '''镜头上移，露出阳光'''
drawpuzzle1() '''躲激光'''
drawpuzzle2() '''对对碰'''
drawpuzzle4() '''数独，同时负责检查是否过关'''
drawpuzzle3() '''迷宫，同时负责检查是否过关'''
```

代码 3：其他处理函数的结构

可以观察到，我们的代码结构是非常清晰的，每一个场景都有对应的处理模块，鼠标和键盘也根据场景的不同做出相应的响应。这样清晰的设计对于游戏的开发和后续的维护和升级提供了极大的便利。

四、后续工作展望

我们的后续工作可以分为五个方向：故事线和世界观的完善、谜题数量和类型的增加、游戏画面和操作的再精致、代码的重构、移植手机。

1. 故事线和世界观的完善

受限于制作时间和展示时间，我们目前只是大概描绘出本作故事的框架，在本作故事内容的广度和深度上，仍然存在大量的可掘挖空间。我们预期本作完整的故事将包含史诗英雄故事全部特征，包括主角的英雄化、持久的旅程、庞大的背景设定、勇敢的行为、神性、大众所知的传统故事。

2. 谜题和玩法的丰富

我们预期大作最终版本将包含数十个或大或小的谜题和尽可能多的谜题类型，并引入动作类的玩法元素，以丰富玩家的游戏体验。

3. 游戏画面和操作的再精致

本作目前实际上并未完全实现水彩风格化，在游戏接口上仍保留粗糙的人物设计，对它们的再精致是我们未来的重点工作之一。另外，为了契合对所设想的谜题和玩法的丰富，在游戏操作多样化的探索，对我们来说也相当重要。

4. 代码的重构

Pygamezero 尽管容易上手，但上限较低，在面对较为复杂的编程时，流畅度可能不如 pygame 本身，因此为了更好地实现我们的设计方案，我们将重新编写代码。

5. 移植手机

本作在悠闲性和思想性的取舍方面作了较好的平衡，相信能够吸引到对故事和解谜感兴趣的人群，本作作为一款成功的单人手机游戏是我们的最终目的。

五、小组分工合作

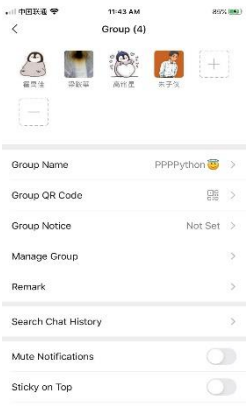
霍禹佳：游戏概念构思、数独谜题设计与实现、视频制作

高铭星：游戏概念构思、背景音乐制作、视频制作

朱子仪：游戏概念构思、所有游戏内的图片绘制、大部分代码实现、实习报告撰写、Poster 制作

梁鞍華：游戏概念构思、剧本、迷宫谜题设计与实现、实习报告撰写

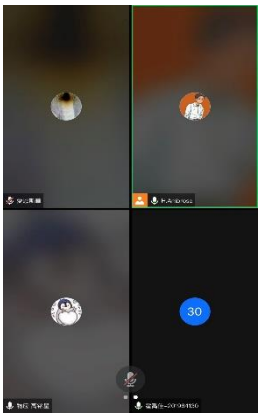
小组讨论情况：

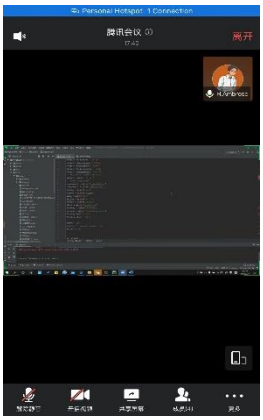

Group (4)
Group Name
PPPPython
Group QR Code
Group Notice
Not Set
Manage Group
Remark
Search Chat History
Mute Notifications
Sticky on Top


PPPPython (4)
线，时间上就很成问题
You recalled a message
我们不如白天开一个视频会看看情况，如果谜题需要减少就减少，讨论完之后看情况，如果一定会超时，我就跟助教商量一下（或者问问他谜题展示能不能压缩时间线缩短）
终极谜题(菜鸡完成任务).rar
Tuesday 1:12 AM
ok！
我试了试！


PPPPython (4)
解谜成功音效
可以 分给咱一点
!
dei
我记一下
Tuesday 1:08 AM
我要不多框架做完了，只剩谜题的部分
但是刚刚试了一次，加上旁白的时间，光走剧情就要1min，我们最后的剪辑还要剪进去挑战失败的支线，时间上就很成问题


PPPPython (4)
只有两种格式可以用
要
这边我留了个后备的
是一个翻卡牌的任务，要是能都加进去我就试试


30


腾讯会议
离开