

廈門大學嘉庚學院

(论文首页纸)

2021—2022 学年第 二 学期

课程名称: Java 程序设计 (电子)

任课教师: 王淑贞

论文题目: 瓦力马尼亚大陆

学号: IST20043

姓名: 黄艺茗

年级: 2020

专业: 智能科学与技术

班级: 20 智能科学与技术 (1) 班

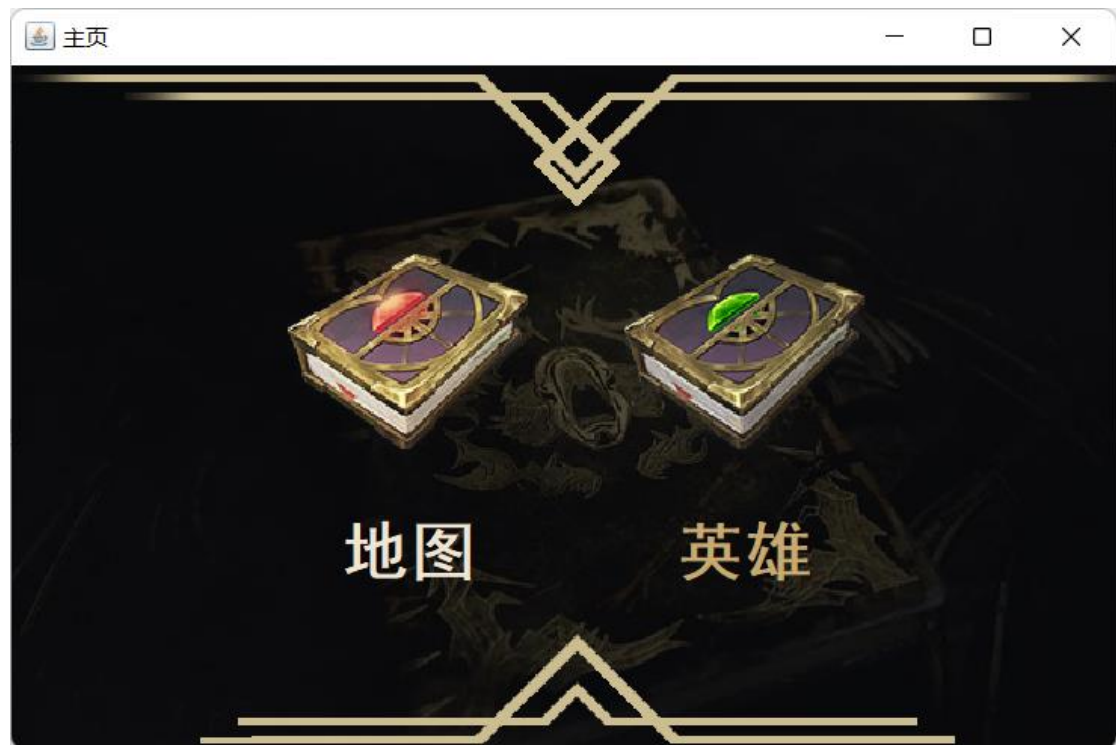
提交日期: 2022 年 5 月 30 日

一、 课题介绍

本次课程设计，制作了原创架空世界观——瓦力马尼亚大陆的世界介绍。其中仅包含原创地图，因为世界观英雄原画与故事不完整，故使用网络资源代替。

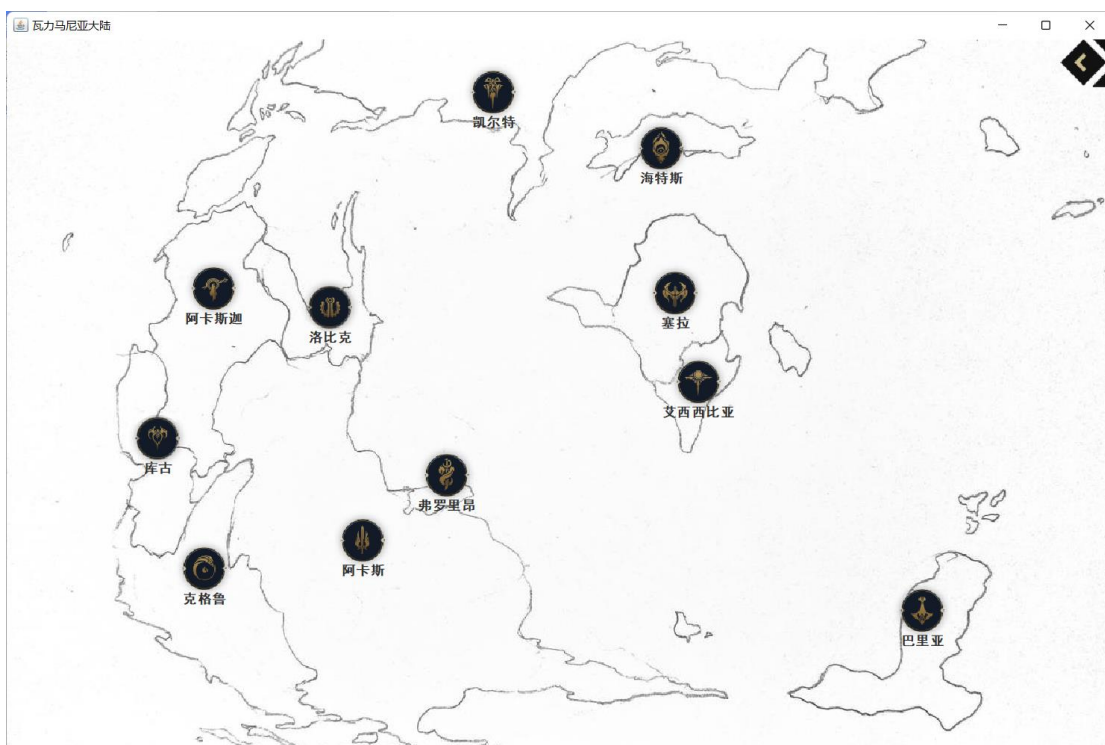
二、 成品操作步骤及结果截图

运行程序，首先出现的是主页。主页是一个提供选择查看内容的页面，可以选择查看地图还是英雄，该页面控制整个程序开启与关闭。



选择地图项，会弹出一个新的窗口，展示地图信息。

地图页的主体是一张地图，根据城邦不同，每个地图上方会有一个 logo；右上方是一个菜单栏，因为现在选中的是地图页，所以地图项高亮，点击右上的按钮可以实现隐藏菜单栏功能。



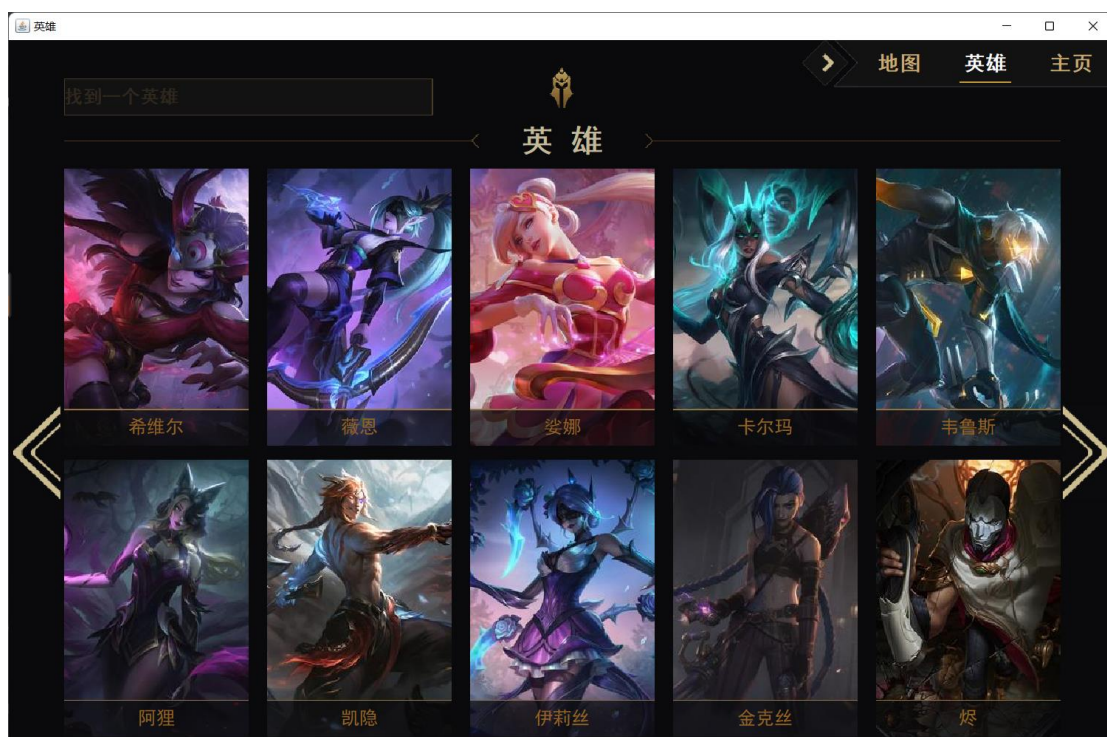
点击任一地图上图标，会在页面内的左端弹出该地区详情页。

详情页中会展示城邦 logo，城邦名称，城邦简述，等等。详情页右端展示了该城邦的英雄代表。打开详情页后，地图上所有图标均会被禁用，防止在打开详情页后，误触导致弹出多个详情页叠加。详情页右上角是一个回退按钮，点击按钮可以关闭该详情页，其他图标被激活。

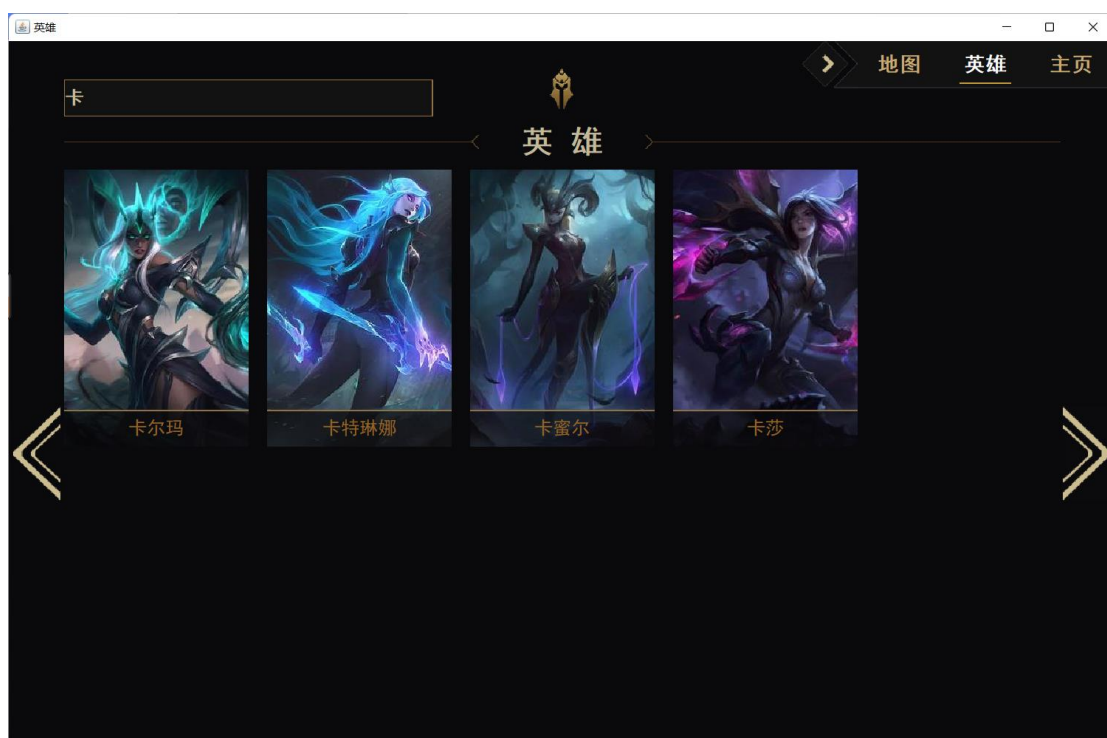


无论选择主页的英雄选项，还是地图页菜单栏的英雄项，都会弹出该英雄预览界面。

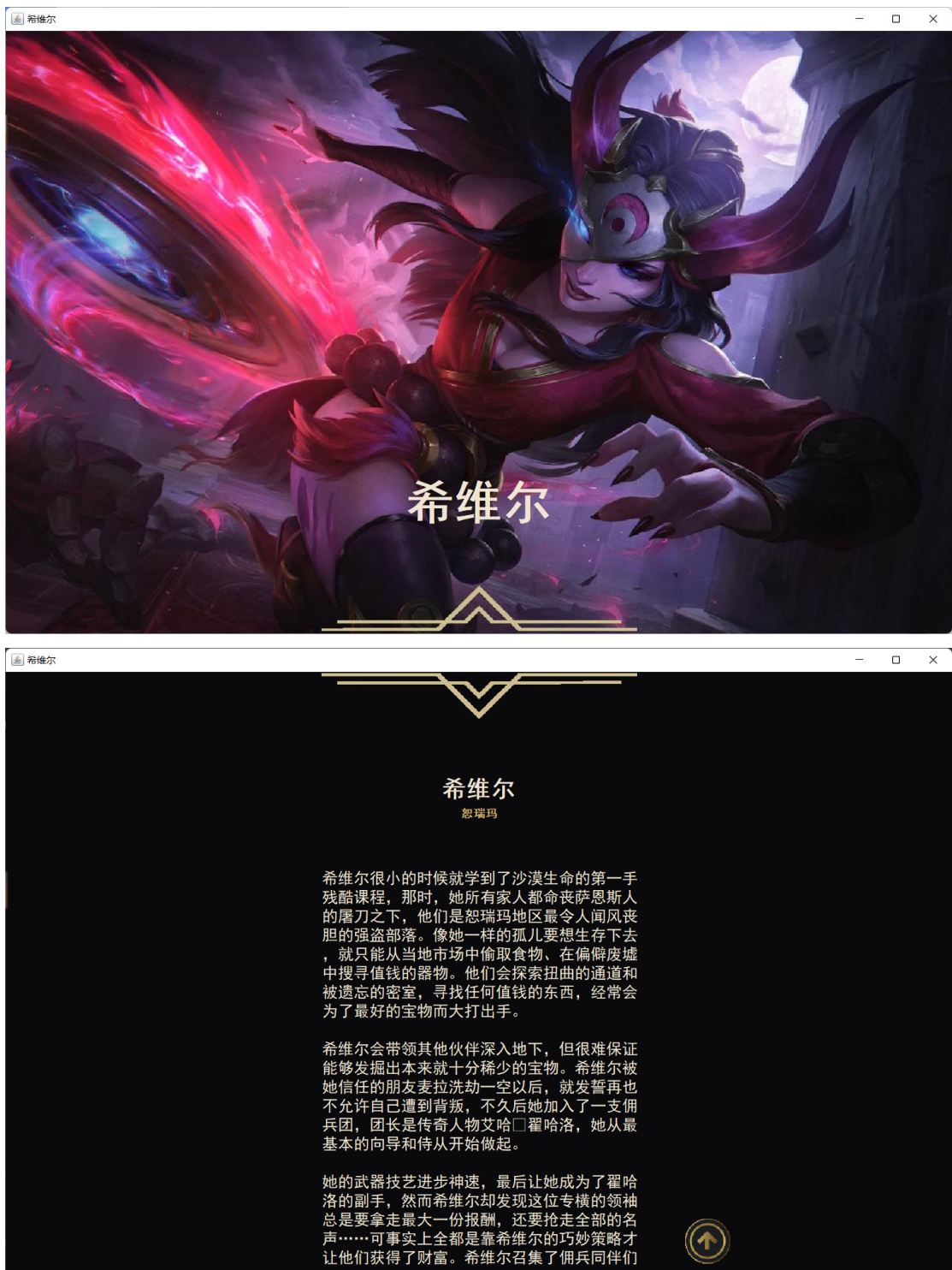
英雄页的主体是十张英雄卡片，每个卡片下方是英雄名，卡片图片是该英雄的图片，点击英雄页两边的左右按钮，可以切换英雄页。



在左上角的搜索框输入时，可以进行对指定英雄进行相关搜索。



点击英雄卡片会弹出英雄详情页，首先是英雄大图预览，点击下方按钮可以开始查看英雄短片故事。使用鼠标滚轮可以慢慢翻页，点击右侧向上按钮，可以将文本重新置顶。



三、 分析与设计

（一）主页

1. 功能设计

主页需要提供的功能只有两个，一个是控制程序的开始与关闭，另一个就是选择进入查看地图，还是英雄。

查看地图和英雄的选择，只需要做两个按钮即可。

2. 画面设计

对于主页的 UI 设计，我选择中心化构图的构图方式，对背景做虚化处理，从而衬托主页的主体内容。主页按钮选择位于页面中心，同时也是视觉中心。主页上下的边框，均指向中心位置，用于将用户视觉重心，拉向中间的主要功。

（二）地图

1. 功能设计

地图页最主要的功能为瓦力马尼亚大陆的地图预览，与各个城邦的版图位置。对于如何查看每个城邦的简介，我选择用按钮来实现点击图标时，弹出城邦简介操作。在按钮的处理上，我对其进行了悬停时，预览城邦 logo 的处理。点击弹出查看城邦简介窗口，从最左方弹出。在描述的右侧主体位置，是一个城邦相关英雄，对该城邦简介进行修饰和补充。城邦简介中内容，除简介内容均为 JLabel。城邦内容数据，从存储的 txt 文件中获取，定位城邦数据，用城邦名称做主键进行绑定。

```
name:巴里亚  
description:与世隔绝的魔幻土地  
governance:无  
technologyLevel:低  
magic:高  
environment:未知  
main:巴里亚是一片与世隔绝的魔幻土地，很少有人去那里，过去的人也几乎没有能够回来的。
```

以上图城邦巴里亚为例，以换行符为分割，提取各项数据。

城邦详情页的右侧透明英雄，我使用了假面板的设计，让用户认为，承载人物的容器更像是人物脚下的底，实际上，在这两者的后方，有一个透明的容器，承载人物的底和人物本身。这样做的优势在于，可以让人物与基底进行绑定，同时出现或是消失。

在按钮的设计上，实际上和上述假背景有异曲同工之处，因为我的按钮中间位置有一个两个三角相减出的区域，为透明区域，直接放在黑色面板容器中，无法表达他的透明感，所以仍然要有一个透明面板容器。但这样做，就会如果黑色背景无法填满区域，而留下按钮下方的一条空白区域。我在解决这个问题时，直接用一无用黑色面白填充了该区域，使整个城邦简介页面成为一个整体。

地图页的左上角，设置菜单栏，用于在该页面开启查看其他页面。为了让用户更好的观看完整地图，菜单栏最左侧设置隐藏菜单栏按钮，点击后可隐藏菜单栏；也可通过再次点击按钮激活菜单栏。菜单栏中该页面的按钮会持续处于高亮，且无法再次通过点击开启第二个同样页面。菜单栏其他项均设置鼠标悬停事件，在鼠标经过时，会高亮处理。

在整个页面布局上，我选择绝对位置布局，但在组件定位时，我仍然用与其他既定位置组件相对位置的方法，决定其定位。这里以比较简单的菜单项举例。

```
menu_base.setLocation(jf.getWidth()-menu_base.getWidth(),0);
```

2. 画面设计

地图页的主体部分就是整个背景地图，所以在查看详情页等等，全部避开了遮挡中心位置的设计，弹出也是从页面侧边弹出。对于画面中的图标设计，全部采用黄色为主体色，黑色为背景色的色调。以右侧图标为例，在图标的设计上，均选择规则形状的拼接与裁剪，尽量不用不规则图形，以为介绍一个架空世界观，实际上是在描述



历史，图形的秩序规则可以强调世界观历史背景的严肃性。

在城邦描述的文字样式上，同样选择笔触横平竖直的黑体，颜色上与画面主体颜色黄色统一，并降低其色彩饱和度和明度。低饱和度，低明度的色彩给人的感觉依然是不活跃的，正适合用在描述一个城邦厚重的历史文化上。

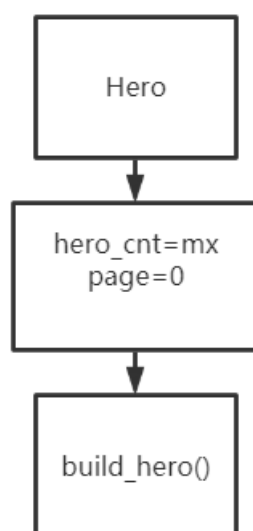
在城邦简介页右侧的人物方面，我使用和左侧背景色相同的颜色的画板作为人物的基地，使一个透明背景的人物，可以立体起来；再者，可以使其从左侧的黑色背景中更加柔和的过渡过来。

（三）英雄

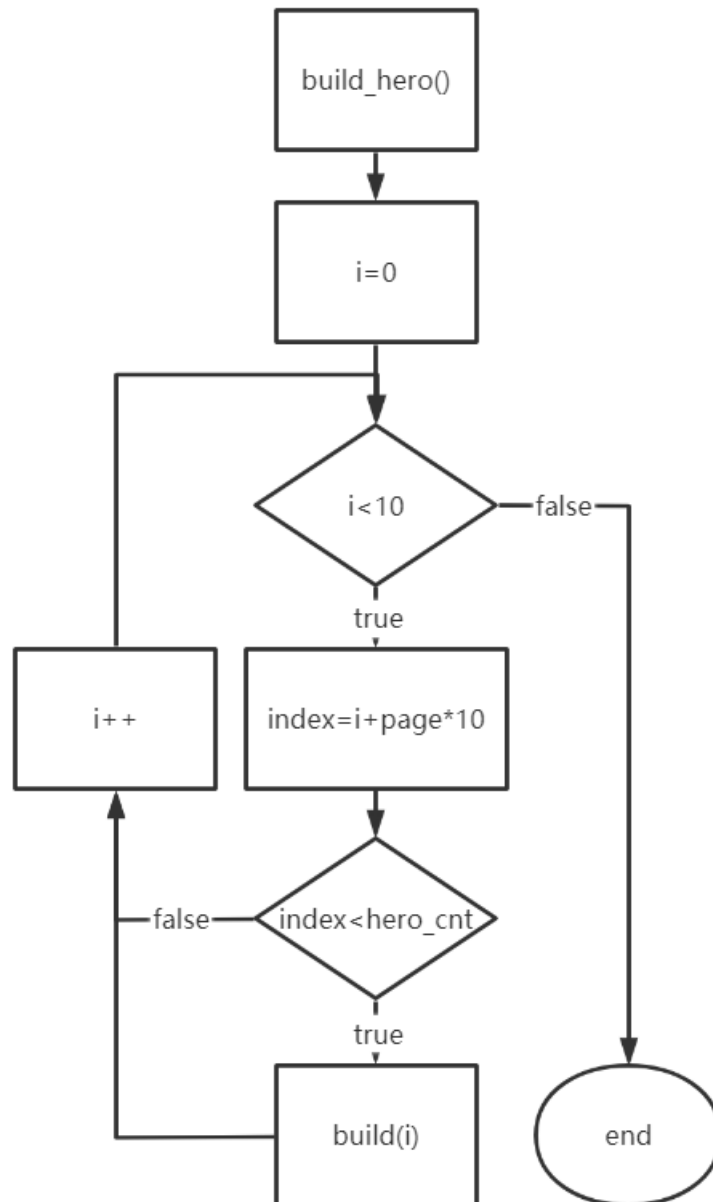
1. 功能设计

英雄页的主要功能为预览马力瓦尼亚大陆的所有英雄。页面主体是十张英雄卡片，点击可以查看英雄背景故事。点击左右两侧的换页按钮，可以进行换页，右上角同样设置与地图页相似的菜单栏。在左上方的文本框中输入文字，可以对指定英雄进行相关查询。

英雄页最主要的模块为绘制十张英雄卡片。十张卡片的实现原理是提前预设了十个固定的卡片面板，然后再每次更改页面时，均只用改变对应面板内的内容即可。卡片为了更好的设计点击查看英雄背景故事功能，使用 `JButton` 组件。为了后续更改卡片的逻辑性与鲁棒性更强，我选择对将要描绘的英雄存入一个数组，并记录其数量，再直接对该数组内的英雄描绘的方式来处理。



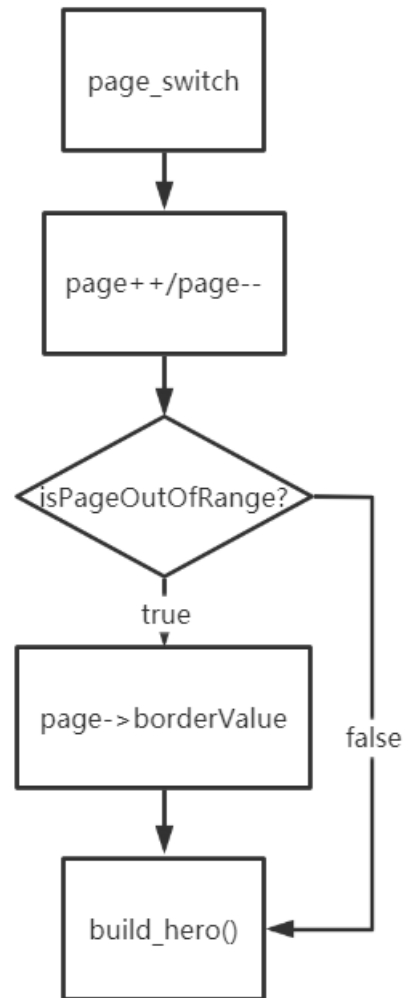
上图为在开启英雄页时，对英雄页的初始化操作。下面将着重介绍 `build_hero()` 模块。



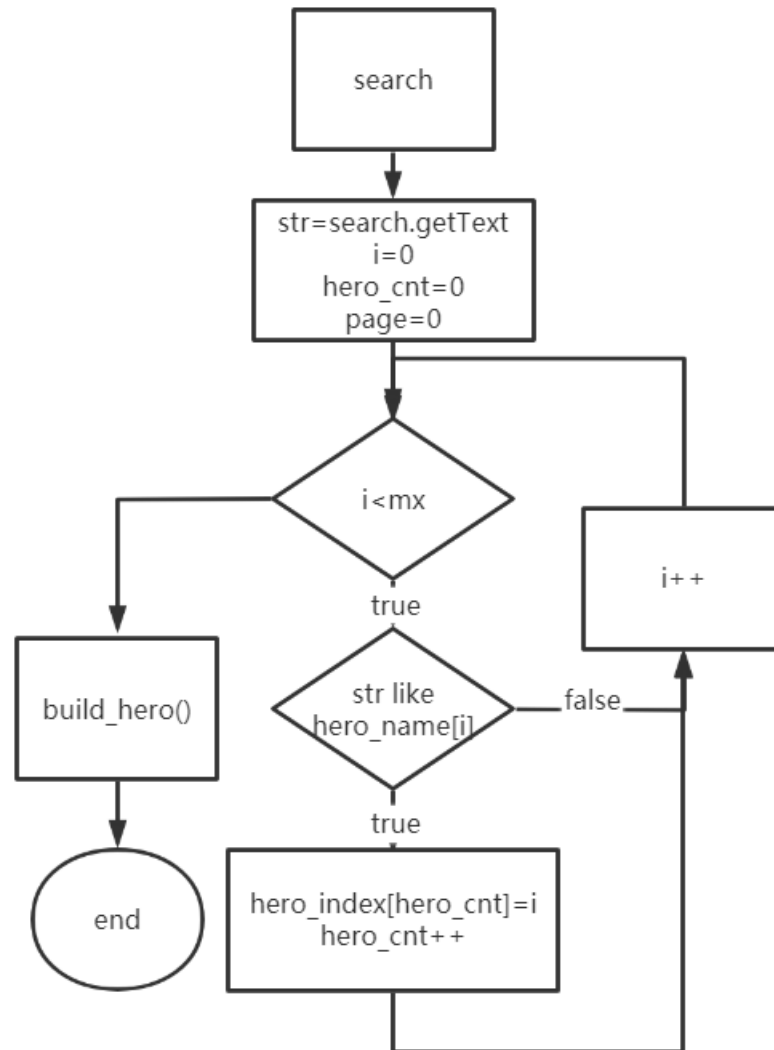
上图为英雄页建立模块关键逻辑流程图，在介绍英雄页建立模块前，首先我已将需要描绘的英雄索引存入数组 `hero_index[]` 中，并将 `page` 页码置回 0。在建立英雄页时，使用 `index` 访问指定英雄。

在建立英雄页时，我会通过当前的 `page` 和构建的面板序号计算得到需要访问的英雄数组下标索引，并通过这个索引，找到真正要构建描述的英雄。我在存储英雄数据时，直接利用序号做索引，所以可以直接通过索引，正确的读入英雄信息，得到他的英雄名和英雄对应插图，并描绘在该卡片内。获取到 `index` 后，还要判断该索引是否已经超过了要描绘的英雄范围，即判断 `index` 是否小于 `hero_cnt`，如果小于，则正常描绘；若大于等于，则直接将该页的面板设置为不可见。最后循环构建十张卡片，结束模块。

在英雄详情页的绑定上，我对十个按钮添加监听事件，为了定位到是点击了哪个英雄，触发弹出英雄详情页事件，我在英雄详情页的构造函数上加了参数，获取从英雄页点击按钮，得到的 `index` 索引值。



上图为换页功能流程图。因为做了上述封装处理后，换页操作逻辑就变得异常简单，只需要在点击左右按钮后使 `page++`或 `page--`，再判断页码是否超过了最大页码，若超过则进行循环处理，最后再重新调用 `build_hero()`函数构建页面即可。



上图为搜索功能的逻辑流程图。实际上搜索功能的本质就是改变即将描述的英雄，即重置一个新的 `hero_name[]` 数组，该数组里的英雄名称均和搜索内容相关。在搜索功能上，我们要做的并不是点击按钮，才开始搜索，而是实时监听文本框中的内容，当他更改时，就开始搜索。当文本内容被修改时，我们首先将 `hero_cnt` 和 `page` 置回 0，因为文本框的修改意味着搜索已经在进行，并且将要迎来一次新的页面构建。之后再判断文本框中是否还有字符串，如果没有字符串，说明不搜索任何内容，直接对页面重新初始化。如果有字符串，那么获取到该字符串，遍历所有的英雄，寻找与其相关的英雄。我在判断英雄是否与文本框中内容相关时，使用直接通过英雄名是否存在文本框中所有字符的方式，如卡蜜与卡蜜尔为相关，但与卡尔玛不相关。在遍历时，没找到一个与文本内容相关的英雄，就将他的索引加入 `hero_index[]` 数组中，并将 `hero_cnt++`。遍历完后，重新调用 `build_hero()` 函数即可仅将相关的英雄构建在英雄页中。

在搜索框中添加默认文字，我并没有使用传统的默认文本内容的方式，而是新建了一个 `JLabel` 组件，将它覆盖在搜索文本框上方，监听文本框内容，当文本框中有内容时，将该 `JLabel` 组件隐藏；否则将它作为默认文本覆盖在上方。

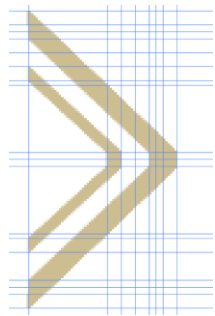
2. 画面设计

英雄页的画面设计，依然保持简约，画面主色调与地图页统一为黄色为主体色，

黑色为背景色，将画面的视觉中心尽可能集中在十个英雄卡片上。在一个英雄获得焦点后，在他的卡片上添加黄色边框，让他能够从十张卡片中突出出来，将用户的视觉可以集中于选中卡片。在英雄名的设计上，我使用了一个半透明的黑色做基底，在与英雄图接壤的上部添加边界线，并用低明度低饱和度的色彩书写英雄名。使用黑色基底可以让该组件与背景更加融合，且不与主体卡片抢夺用户视线。半透明则可以让该组件过渡到卡片主体时更加自然。低明度低饱和度的英雄名，同样也起到不抢夺主视线的作用，让用户能更轻松的聚焦于英雄卡片主体。

搜索框的设计上，我并没有用任何的外部提示信息，告诉用户可以用该文本框来搜索英雄，所有就一定要在文本框中添加默认的文字，来提醒用户，这边可以用于搜索。但默认文字又不是主要信息，故我将其明度饱和度再次降低，使他的视觉重要程度再次降低。而在该文本框失去焦点时，该文本框的边框也将变得更暗。当文本框获得焦点，文本框的边框便会变成激活状态，色彩明度饱和度和度升高，让用户能更好的聚焦于输入内容。在没有输入文字时，为了更贴合默认文本，光标颜色同样被设置的更加暗淡；但当文本框中存在文本内容时，光标颜色便会恢复应有的光泽，这同样也是为了适应其输入文本的颜色。

在图标的设计上，大量使用了参考线，进行规范化处理，使最后画出的图标可以更加有序整齐。这样对标准几何体进行空心处理的方法，可以使图标既不欠缺严肃属性，又不缺创意性。



（四）英雄详情

1. 功能设计

英雄详情页面由两部分组成。该页面的主要功能就是描述英雄的背景故事信息。首先详情页面的第一部分是该英雄的完整插图，该插图覆盖整个页面，页面中下方是该英雄的名字。正下方是一个隐藏插图按钮，点击后，隐藏插图，解锁详情页的第二部分，背景故事部分。该部分主体由黑色背景和中间占大约 1/3 页面宽度的背景故事栏组成。使用鼠标滚轮可以实现故事文本向下翻阅，点击右下按钮，可以实现重新置顶文本操作。再次点击正上方按钮，可以拉回插图，重新覆盖页面。

英雄详情页的构建需要一个 index 索引参数，根据参数找到对应的英雄数据，并提取出英雄的数据，对页面进行构建。

为了区分英雄详情页的两个部分，我设置了两个基础面板容器，大小均为页面大小，通过点击按钮，使二者交替出现，实际为控制面板是否可见属性。在插画页，插画直接覆盖面板，点击正下方按钮，设置该页面为不可见，并让背景故事页设置为可见。

背景故事页中，我使用了 JScrollPane 组件，为了可以隐藏 JScrollPane 组件中的滚轮，且还不禁用他的功能，我将 JScrollPane 的大小设置为超出页面一些，将滚轮挤出页面，达到隐藏效果。为了让文本框只在页面中间，我将一个面板容器放入 JScrollPane，并让两个和文本框同宽的标签组件与文本框一并放入面板容器，并均分排列。并在初始化完页面后，将文本置回顶端。并在右侧设置一个置顶按钮，用于更便捷的人为操作文本置顶。

2. 画面设计

在英雄详情页被打开后，映入眼帘的便是该英雄的插画覆盖整个页面，这本身就是一个简单，但是亮眼的页面设计，页面中下方英雄名以低饱和度，相对高明度的色彩弹出，这种低饱和度的色彩，不会抢夺视线，并且高明度的色彩又不会使用户观看时非常费力。

收起插画，打开英雄背景故事页，英雄名为标题伫立在中上方，名称下方为地区信息，这里的地区信息使用的是低饱和度低明度的色彩，恰好适应的是它内容重要性不高的角色，反衬标题的高明度。正文部分不使用加粗字体，在长文本中，更利于用户观看文字。

黑色背景亮色正文，加上中心式构图的方法，可以让用户自然地将视线聚焦于文本内容，观看时可以获得更好的体验。人的最舒适的视线左右夹角为 120° 之内，实际上，让文本只存在于页面正中，同样也出于这个考虑。

四、 遇到的问题和解决方法

数据：

在数据的存储方面，我有许多可以选择的方案，当然最开始，也是最常见的，我会想用 MySQL 数据库来存储数据，这显然是一个很有效的方法；但在我目前使用的小规模数据中，我认为无需使用，而是直接将数据存入 txt 文件，用换行符来分割数据。这样做很大程度上，减轻了我的工作量，让我不用对连接数据库进行额外的学习。

地图页：

在地图页中，点击 logo 查看城邦简介时，依然可以点击其他按钮，弹出新的城邦简介页，如此，会产生左侧叠加许多简介页现象，甚至会因为误触，而覆盖用户当前查看的城邦简介页。我用禁用按钮解决了这个问题。当打开一个城邦简介页时，将所有按钮置为禁用状态；当关闭城邦简介页时，重新激活所有按钮。

英雄页：

在英雄页中，英雄卡片上的英雄名称显示，是一个问题。将英雄名称作为一个标签组件，调整 z 轴，覆盖在按钮上时，再次点击按钮，会使按钮获得焦点，重新覆盖在英雄名称上。这个问题，如果按钮和标签在同一组件时，我无法解决。我用让两个组件位于不同图层的方法，让按钮的焦点获得，无法影响标签组件的显示。我在十个卡片中的底部，额外生成十个英雄名称面板，将英雄名称放入这个面板中，解决了这个问题。

在生成英雄卡片，构建英雄页上，渲染速度非常慢。我在排查问题时，发现是因为，我在处理图片数据时，没有做到完全处理，而是在渲染时，实时进行图片的等比例缩放，而我的图片又是清晰度很高的高质量插图，在每次的页面渲染时，都要进行十个卡片图片的等比例缩放，这无疑是一个效率非常低，速度非常慢的方式。我在处理这个问题方面，选择直接对数据进行优化。我用 python 对我的图片数据进行了批量缩放，缩放到它本应该是的大小。这样，让我在渲染页面时，无需在额外处理图片的缩放工作，页面的渲染速度明显加快。

在建立英雄卡片的函数逻辑上，是否可以做到最泛用，也是一个问题。最开始，我利用纯页码和卡片序号计算英雄编号，进行定位。但如此做，在搜索后，重构英雄页时，就产生了很大问题，搜索重构无法调用建立英雄卡片函数，需要重新写一个方法，构建页面。为了解决这个问题，我定义了英雄索引数组，让它存储将要描绘的英雄的索引，在遍历重构页面时，直接按照存储的英雄索引，构建页面。有了这个英雄索引数组，搜索的本质就是通过匹配字符串，改变索引数组。如此，就可以让所有的需要重构页面的模块，直接调用建立英雄卡片函数。

英雄详情页：

在英雄背景故事的文本信息构建上，使用 JScrollPane 组件时，如果启用垂直滚轮，会有一个非常不协调的滚轮。为了解决该滚轮问题，我将该组件宽度放大，用页面大小限制组件的显示，从最右方隐藏了滚轮。另一方面，如何让文本信息处于 JScrollPane 的正中间也成为问题，我的解决方案是将一个面板容器放入 JScrollPane，再将两个与文

本框同等大小的标签组件与文本框一同放入面板容器，并且宽度三等分面板容器，旁边两个标签的背景色设置为透明或是与背景相同色即可。

五、 总结

本次 JAVA 课程设计，我选择了自选题目，进行创作。其实在最开始我认为这份课程设计的难度系数可能会有一定含金量，但最终做完后，回顾整个的功能，好像体现不出什么技术含量。也许是因为我对画面设计花了不少功夫。在一方面，完成这样一个作品，对于我自己来说，本身就是一件很开心的事情。

这次的课设我选择做了一个原创架空世界观的是介绍。虽然说世界观数据还并不全面，但是我仍然可以先用网络资源，对其进行一定程度上的填充。在完成这次课设时，我有在一定程度上，感受到之前的学习给我带来的经验，比如在解决英雄卡片构建时，我借用了上个学期做 python 课设的经验，沿用了之前使用的方法。在我的课题完成过程中，我同样用到了许多老师没有讲到的知识，想要解决这类问题，无疑是要自学的。同样的，我也碰到了一些我无法快速学习，并很好的解决的问题。在处理这种问题上，我只能退而求其次，寻找更简单的解决方案，来实现看似相同或是类似的效果。

同样的，通过完成这次 Java 课设，我对 java 图形界面的了解更加深入，在编写代码时，也更加得心应手。从最开始的需要不断翻阅资料，后面再解决问题时，就是直接熟练的几行敲下来。在数据存储上，我也是第一次尝试用 txt 存储数据，真正用下来，我觉得其实并不难用，倒不如说，txt 存储数据的方式，学习成本很低，对于小规模的数据来说，应该是非常合适的。在书写代码时，代码的逻辑是非常关键的，逻辑清晰，模块高内聚，低耦合的代码是非常好调用，修改以及维护的。在自己梳理时，也会清晰很多。其实在我自己做课设的过程中，学到了许多老师没有教到的内容，比如对于 Java 图形页面的结构，在我进行页面布局时，我的背景总会加在 layout 层，并且直接加在 layout 层，再用绝对 index 值控制 z 轴添加其他固定的基底面板，这样的方式解决了我很多的问题，虽然很好用，但我觉得也许会有什么潜在问题，我暂时还无法发现，但是就目前来看，这种方法确实是有它的可取之处。

这次课设，还有一点是我非常满意的。就是我第一次自己绘制 UI 界面的图标。虽然图标的设计，参考了其他网站，但作为第一次手动从零绘制图标，我认为还是很成功的。在绘制图标时，我用了许多参考线来定位节点，让我绘制的图标，几何形态更加优秀，对称性更好。在各种组件的布局方面，我做的也让自己比较满意。

另外，这次的 Java 课设做的题目，是有关于我自己的原创世界观的世界描述。这个世界观是我从初二开始创作的，到现在为止，一直没有一个很好的展现形式。这次做了这个题目，我就可以在介绍它时，直接拿出这次的 Java 课设。另外，我下学期报名了 vr 实验室，在 vr 实验室做的游戏，会以我的世界观作为背景，明年我就可以直接用这个来和我的队友介绍瓦力马尼亚大陆，这显然是更形象生动的描述方式。

六、 参考资料

官方文档

Java 课件

https://blog.csdn.net/weixin_43924623/article/details/89196561?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522165383668416782248527081%2522%252C%2522scm%2522%253A%25220140713.130102334..%2522%257D&request_id=165383668416782248527081&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-4-89196561-null-null.142^v11^control,157^v12^cont

rol&utm_term=java+%E5%9B%BE%E5%BD%A2%E7%95%8C%E9%9D%A2&spm=1018.2226.3001.4187