

设计

1 前台：界面与交互

1.1 登录界面

登录界面用于为用户提供输入用户名及密码的界面。

界面上有两个输入框：用户名输入框，密码输入框

两个按钮：登录，注册

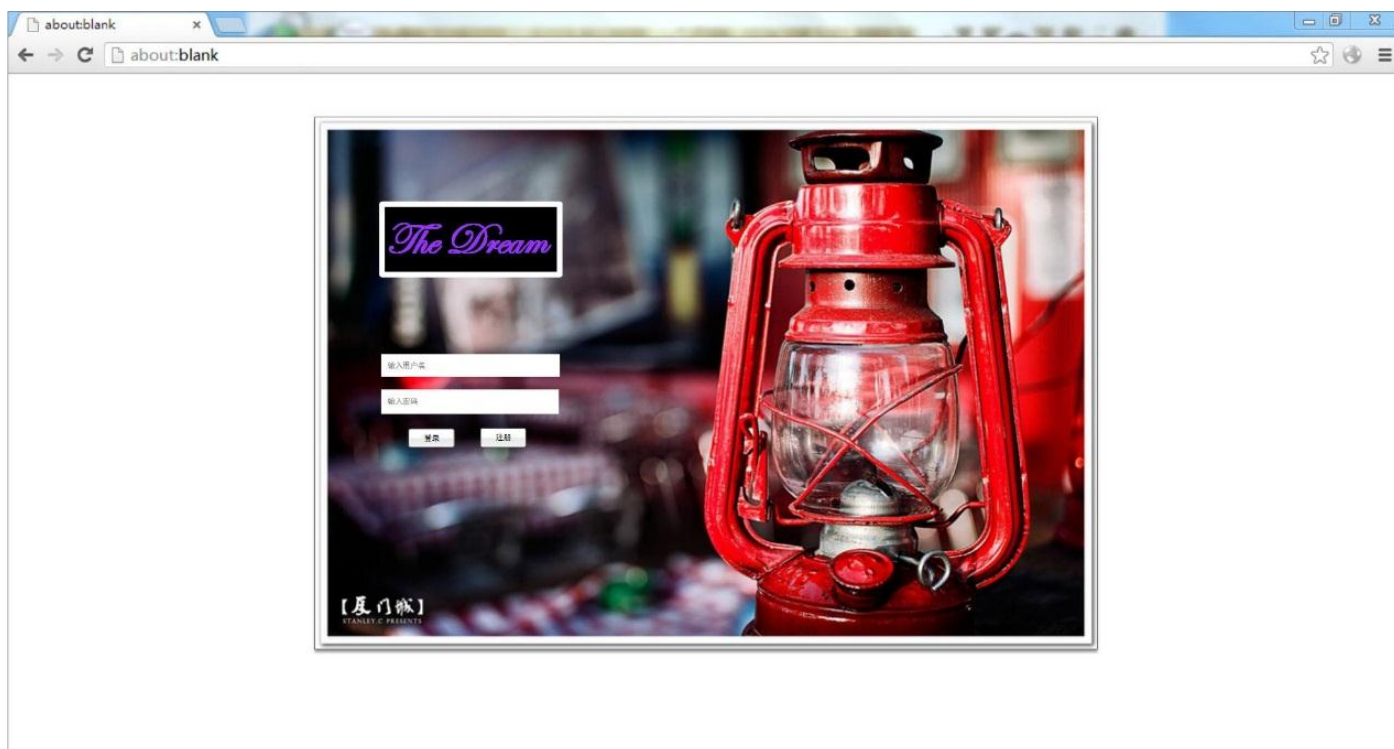


图 1. 登陆界面

在登录界面输入好用户名和密码之后，如果符合规范，单击登录将会进入开始游戏界面。如果用户没有注册过，单击注册，跳转至注册界面。

1.2 注册界面

界面显示“用户注册”，有三个输入框：用户名，密码，邮箱（可在图中多增加一项确认密码）。下方有个按钮：“I am ready for being a dream maker!”。



图 2.用户注册界面

在注册界面完成相应信息填写后，如果符合规范，单击“I am ready for being a dream maker!”按钮，将会跳转至新手模式界面。新手模式不断单击下一页，至到尾页（此时假设用户已经对游戏的操作有了一定的理解），点击“进入游戏”，跳转至开始游戏界面。

1.3 开始游戏界面

开始游戏界面有四个按钮，上面两个按钮显示任务模式和自由模式，下面为游戏设置和新手教程。在游戏设置的弹出框里，用户可以设置音乐的开关（单选框：MUSIC-ON/MUSIC-OFF），以及可以选择“注销用户”按钮。

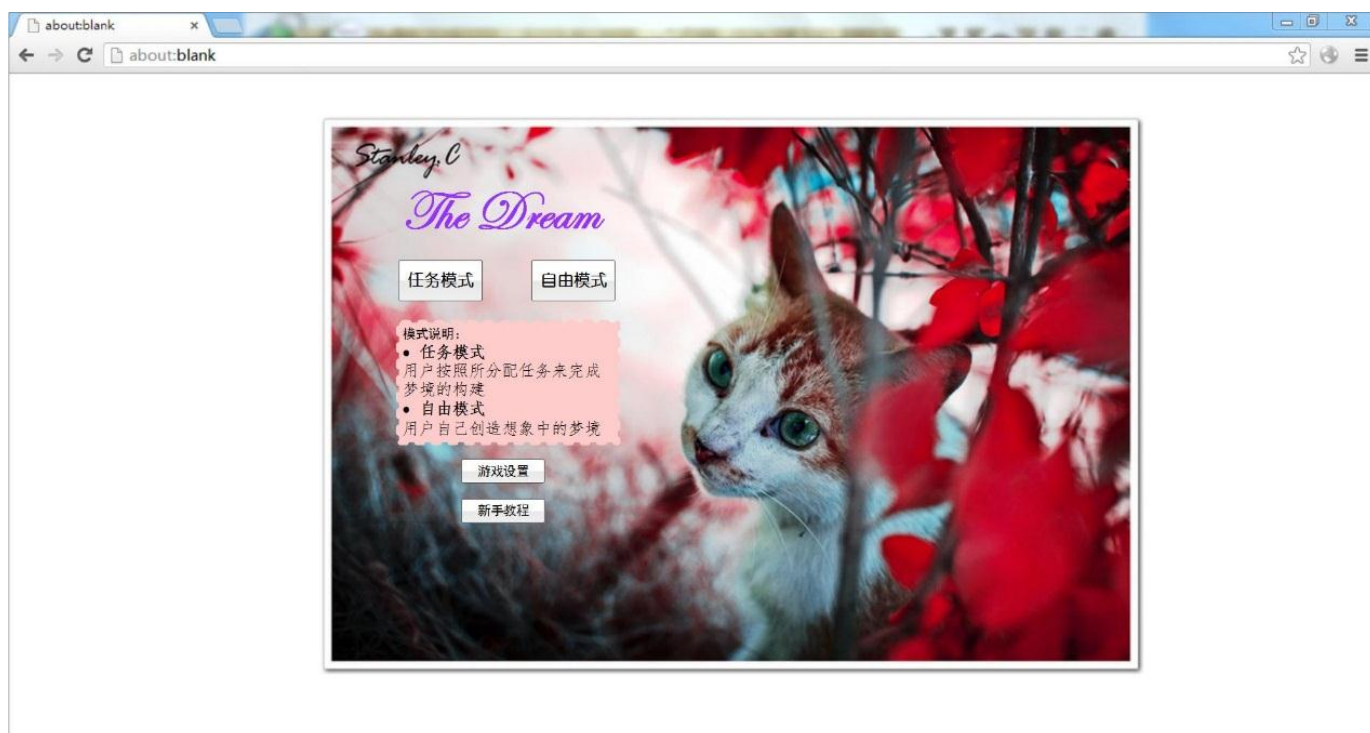


图 3.开始游戏界面

用户点击任务模式，将跳转至任务模式界面，点击自由模式，将进入自由模式界面。

用户单击设置游戏，将出现弹出框，框内选择 MUSIC-ON 没有操作，MUSIC-OFF 将关闭音乐文件。单击注销用户将跳转到登录界面。

1.4 新手模式界面

新手模式界面（即新手教程）将给出游戏的操作的方法。由图片（或者视频）组成。教程结束时出现按钮“进入游戏”。

1.5 自由模式界面

①进入自由模式后，首先弹出的是一个欢迎界面，如图 1，在这个对话框中我们有多个场景模版可供选择，任意选择一个，单击“下一步”继续操作。



图 4. Welcome Dialog Box

②成功选择一个模版后，The Dream 会进入游戏界面，并在左上角显示模版场景，如图 4，5 所示。

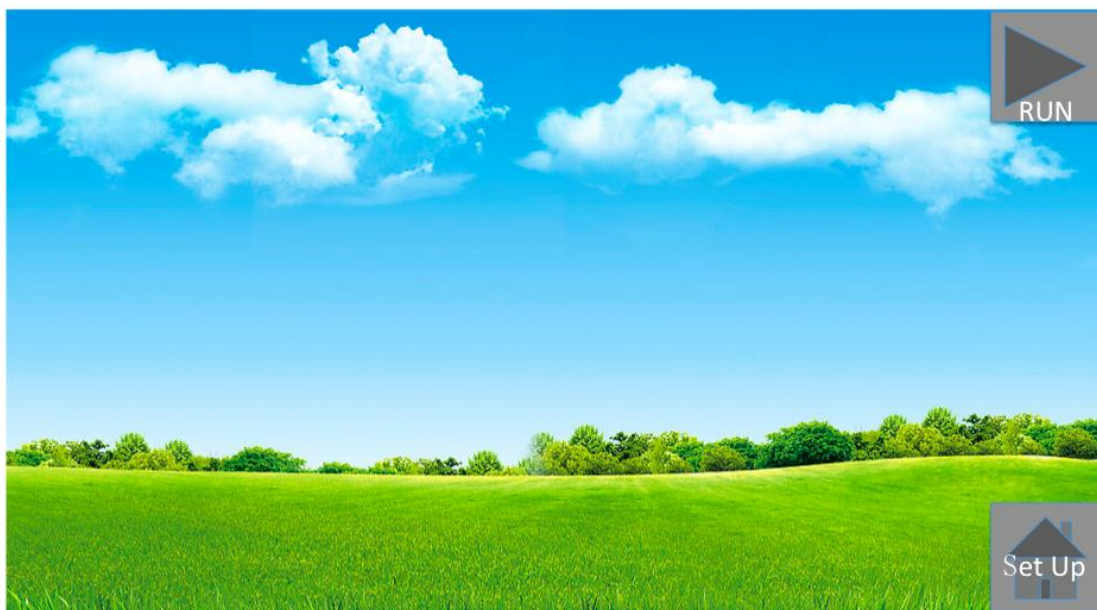


图 5. Scene View

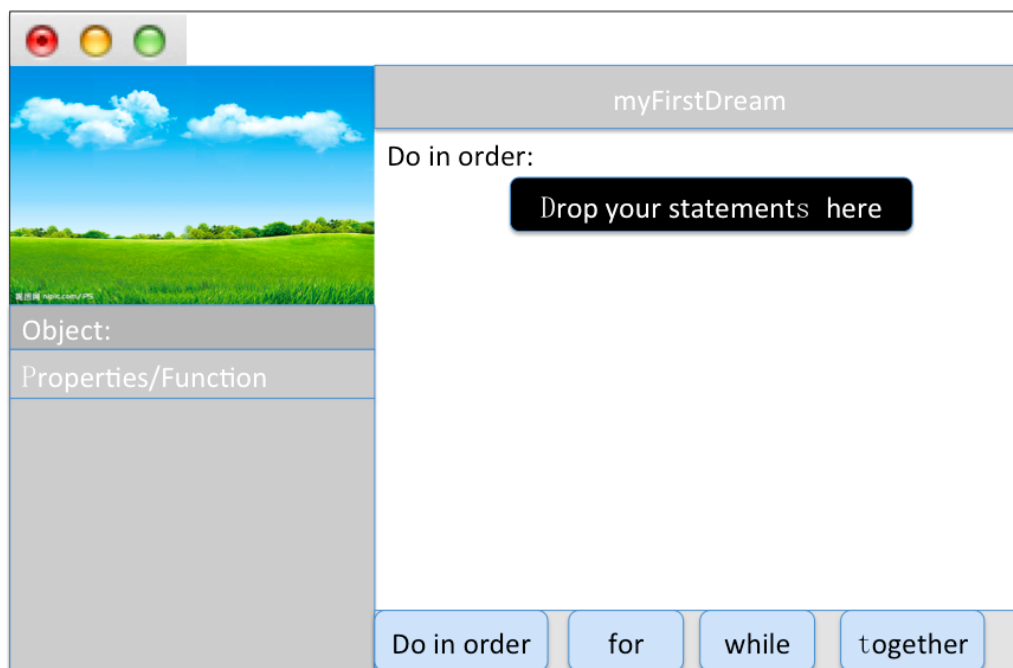


图 6. Start scene, Code Editor

这里，我们把这个 Start scene 分成四个 panel：左上角为预览窗 Scene View；左下角为 Method Panel，记录了 object 的属性和方法；右上部分为 Code Editor Panel；右下部分为 Control Panel，控制方法的运行顺序。

③点击 Scene View 里的 Set Up 按钮，就可以进入场景编辑页面，如图 7。场景编辑页面由两个 panel 组成：Scene Setup Panel 和 Gallery Panel。这个编辑器的主要目的就是通过添加、摆放 Object 来构造一个虚拟世界。在 Gallery 里包含了许多物体的模型，我们这里按照继承关系把物体分为了五类：Biped Class（两足类），飞行类（Flyer Class），Prop Class（静物类），Quadruped Class（四足类），Swimmer Class（水生类）。用户可以

选择自己想要的东西拖到画布上。

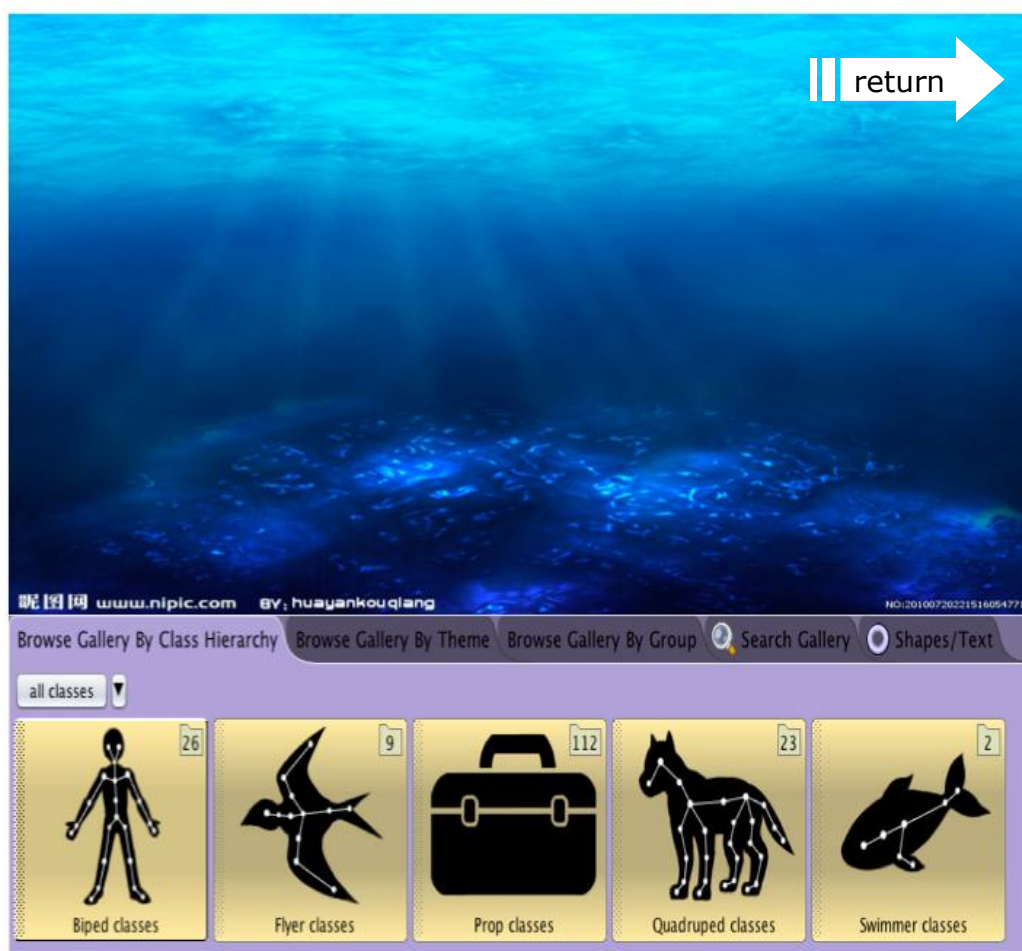


图 7. Scene Editor

选择 gallery 里的一个文件夹，比如选择了鱼类，则出现各种鱼可供选择，如图 8.

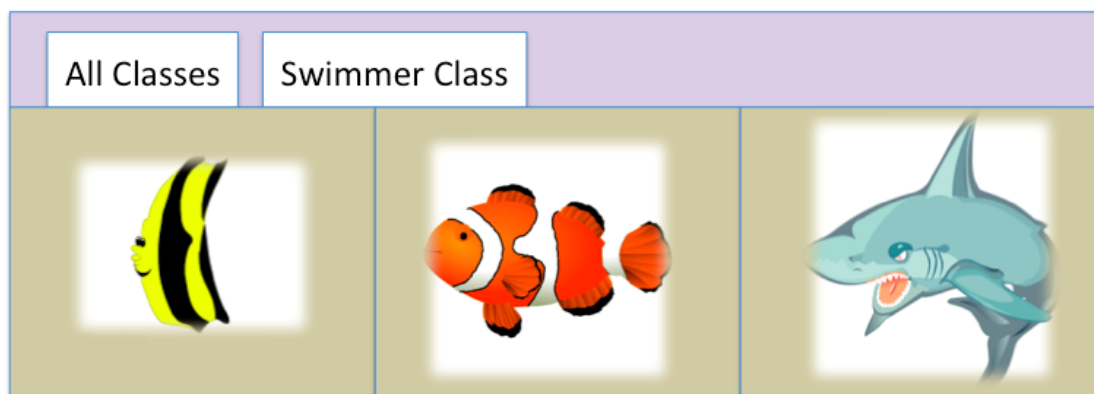


图 8. Swimmer Class

从这些图中选出自己需要的那个，单击即 new 出了一个 object，系统弹出对话框，要求玩家给新建的对象赋予属性。如图 9 所示。赋值完成后点击“ok”确认完成，新建的对象被添加到场景里面，如图 10 所示：

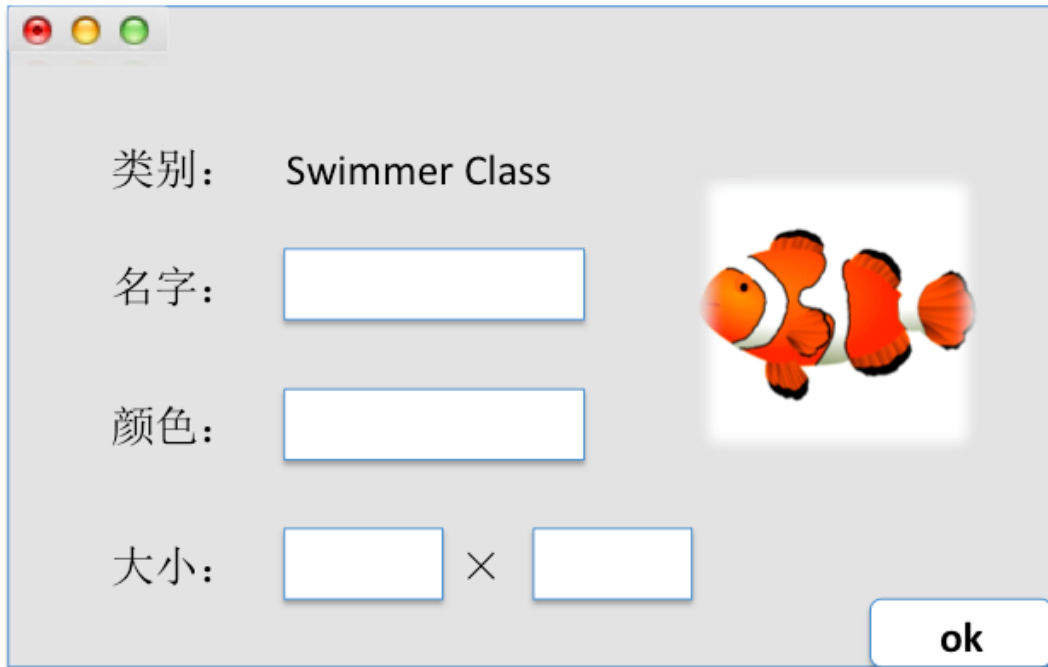


图 9. Assign values to properties

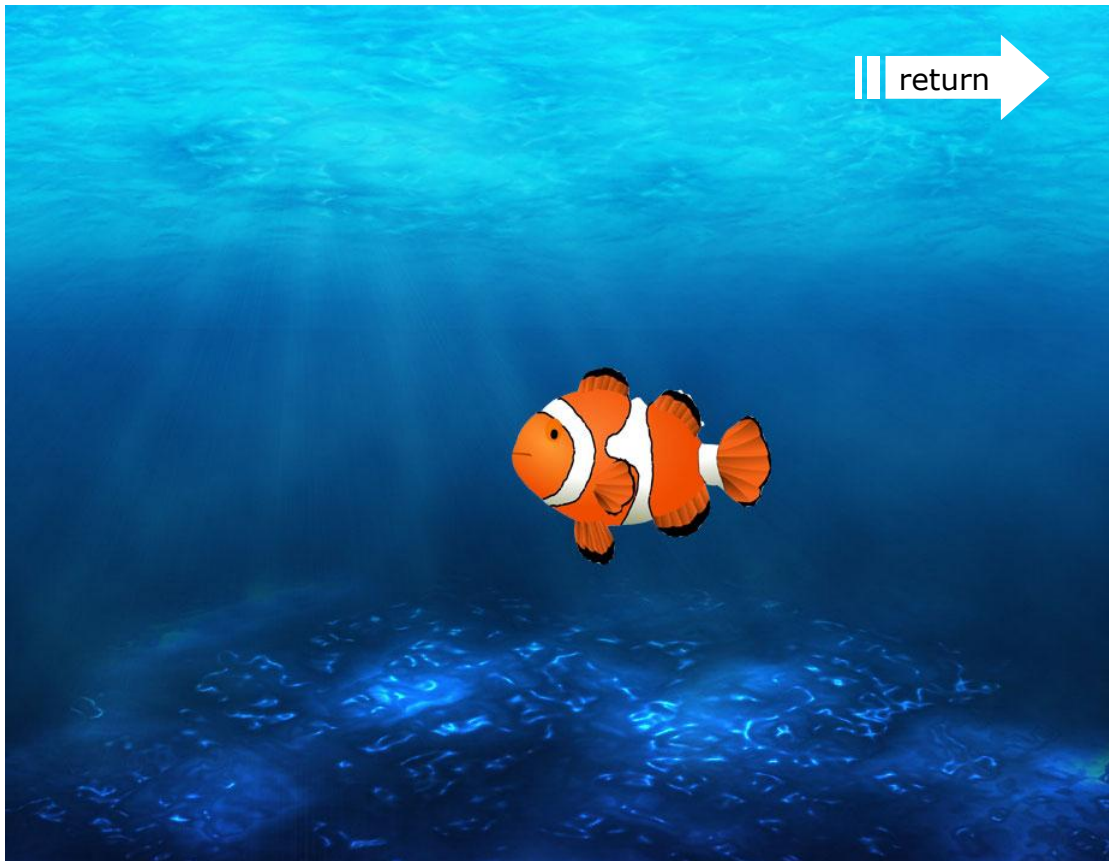


图 10 把小鱼加入场景后的合成图

类似的，用户可以根据自己的喜好再加一些别的东西上去，假设完成后如图 11。

完成 Scene Edit 之后，单击 return 返回 Start Scene Code Editor 界面继续编辑。用户创建的对象都会在实例列表中被列出来，用户可以下拉实例列表选择对象编辑。



图 11.选择 object 进行编辑

选定 object 后，在 Scene View 下方的 Method Panel 里有一系列的项目，每个项目都代表了该对象能进行的操作，或者代表了该对象的祖先，以便于展示继承关系。如图 9 所示。游戏 The Dream 中的每个类都预设了一些方法，包括：说、想、move、moveToward、moveAwayFrom、moveUp、moveDown 等等。用户还可以根据这些方法组建新的更复杂的动作，如跑、跳等。最后在 Code Editor Panel 中按照逻辑顺序，添加方法。完成后点击 Scene View 里的 Run，就可以看到程序运行的效果了。



图 12. 小丑鱼的部分方法

1.6 任务模式界面

用户通过点击左上角的“任务”按钮，来查看文字版的任务类型。



图 13.任务模式

点击“主页”按钮回到开始游戏界面，放弃该次任务。

点击“个人信息”按钮，弹出框显示用户的个人信息，包括用户名，邮箱，现有金钱，等级。

点击“商店”按钮，弹出框里面为商店界面。

可以单击对象框旁的“新建”按钮来创建新对象。单击某个 Object 来确定选中的对象。

在属性栏中，用户可以通过选择属性的某个状态来实现任务中的要求（通过单选框或者下拉框）以及函数框中的某个函数来完成动作，时序结果在右下方显示，可视化结果在左上方的云雾中显示。

（具体实现类似于自由模式，于此不累述）

1.7 商店界面



图 14.商店界面

显示现有的金币以及商店里的技能。如果用户一开始没有颜色这个属性，那么要先用一定的金币来购买才能使得这个属性被开启。当拥有了颜色这个属性之后，就可以在颜色里面（会有相应的下拉框中）选中购买相应的颜色点击购买。可以实现往用户个人的属性库中增加属性的功能。

1.8 用户信息界面

显示用户名、用户邮箱、用户等级、用户现有金币、用户现有的属性（比如，在 COLOR 一栏中已经购买了红色、蓝色、绿色）以及行为（即函数）。

2 后台：

2.1 业务逻辑和数据流程

2.1.1 登录

在登录页面中包含判断输入规范的 JS 文件。用户名输入过程中，判断用户名的长度是不是 4-16 个字符以及必须由数字和字母组成，在输入框的后面显示状态。。输入密码的过程中，判断密码长度是否在 6-10 位。当上述两项有任意一项在输入规范方面不满足，则当单击登录的时候，提示：“请输入正确的用户名或密码！”。如果符合规范，则将“用户名-密码”与数据库中的“账户名-密码-邮箱 TABLE”比对，如果不存在这样的表项，则弹出提示“用户名不存在或者密码错误”，保留在当前页面并且清空密码的输入框。如果确实存在这样的表项，则跳转至开始游戏界面，并且载入数据库中相应的用户信息（账户名、总金钱、等级、已完成任务、现有金钱、现有属性库）。

当用户单击“注册”时，跳转到注册页面。

2.1.2 注册

在注册页面中包含判断输入规范的 JS 文件。用户名 4-16 字符，由字母和数字组成；密码 6-10 字符；邮箱应该符合邮箱的规范（若有确认密码，则要与密码的输入框中的内容一致）。输入框后方显示输入是否符合规范的状态。当用户单击“I am ready for being a dream maker!”时，如果有任何一项不符合规范，则显示“输入不满足规范~”。否则，将用户名与数据库中的“账户名-密码-邮箱 TABLE”中的账户名项比对，如果账户名已经存在，则弹出提示框：“账户名已存在，请更换用户名”，并清空用户名的输入框；若不存在，则跳转到新手教程，并且在数据中的“账户名-密码-邮箱 TABLE”以及“账户名-账户总金钱-账户现有金钱-账户等级-已完成任务-账户属性库编号 TABLE”创建新的表项，初始化，并分配一个尚未被使用的账户属性库编号。以及初始化一张“属性的拥有状态 TABLE”。

2.1.3 自由模式

用户进入自由模式后，选择场景，将场景的 ID 传给显示 Scene View 的元素，Scene View 上显示场景。点击 Scene View 进入场景编辑页面，上部分为 Scene Setup Panel，显示用户选择的场景，下部分调用用户私有属性库，将具有的属性和函数列举在下方。

用户选择一个 object 并完成弹出框的属性后，将信息传给显示 Scene View 的元素，Scene View 上将显示相应的对象。

完成 Scene Edit 后，单击 return 回到 Start Scene Code Editor 界面，用户单击下拉键，显示 Scene View 的元素中保存的对象——罗列。用户继续添加动作。完成后单击 Scene View 中的 RUN 按钮，这些信息会传给显示 Scene View 的元素，完整的梦境将在 Scene View 上显示。

2.1.4 任务模式

类似于自由模式的处理方法。

但是当刚刚进去任务模式时，我们根据数据库中的“账户名-账户总金钱-账户现有金钱-账户等级-已完成任务-账户属性库编号 TABLE”得知用户的等级以及已经做过的任务编号，并且根据等级来查看数据库中“等级-任务-奖励金钱数 TABLE”来随机选中相应等级的任务，排除掉已经做过的任务（如果已经全部都做过，就随机产生）。并且把任务的文字描述显示给用户。

之后的处理类似于自由模式，但是当点击 DONE!的时候，将根据用户给与的梦境以及任务的契合度来给与一个百分比，百分比乘以这个任务的金钱数便是用户获得的金钱，在“账户名-账户总金钱-账户现有金钱-账户等级-已完成任务-账户属性库编号 TABLE”中更新账户总金钱，账户现有金钱，账户等级，已完成任务。

2.1.5 购买道具

在购买道具页面，调用数据库中的“商店属性库 TABLE”，把所有的属性以及该属性的状态，和控制对象行为的函数一一列举，并显示对应金钱数。当用户选中一个时并单击“购买”按钮时，先检查用户现有金钱数是否超过该物品的金钱（没有超过则弹出显示“余额不足！”），若超过，则遍历用户私有的属性库，检查该物品是否已经存在（若存在则弹出显示“已经拥有该属性/该状态/该函数”），如果不存在，则提示“购买成功”，在“账户私有属性库”中添加相应属性，账户现有金钱数扣去相应数目，更新“账户名-账户总金钱-账户现有金钱-账户等级-已完成任务-账户属性库编号 TABLE”。

2.2 数据库维护（大概）：

（运用 SQL Server 来构建数据库）

2.2.1 账户名-密码-邮箱-USERIDTABLE

保存用户的账户名，对应的密码和邮箱以及用户的 ID。

2.2.2 账户名-账户总金钱-账户现有金钱-账户等级-已完成任务-账户属性库编-ACCOUNTID 号 TABLE

保存用户的账户名，账户总的金钱数目，账户现有的金钱，账户等级（根据账户的总金钱数来定），已完成的任务编号，账户属性库编号以及 AccountID。

2.2.3 账户私有属性库：账户属性库编号-属性的拥有状态 TABLE

根据账户属性库编号，调出具有该编号的表项，表项中记录账户拥有哪些属性、函数，以及属性中的哪些状态。

2.2.4 等级-任务-奖励金钱数-TASKIDTABLE

记录在什么等级对应的任务是什么，以及完成任务相应得到的金钱数。

- 场景：（括号里为场景中固有的生物场景，不可设置）

- 一 大海里（可以有很多鱼类在游）
- 二 草地上（陆生动物或者人）
- 三 雪地里（陆生动物或者人）
- 四 天空中（鸟，云朵）
- 五 教室里（人）
- 六 沙滩上（人和陆生动物，海龟之类的其他）

- 任务：（以大海场景的任务为例子）

对象及其属性：

海洋生物：宽度（width） 高度（height） X 坐标(xcoordinate) Y 坐标 (ycoordinate) 颜色(color)

动物：

鱼	鲸	龟
---	---	---

植物：

海藻	海葵	海星
----	----	----

任务 1:

目的：规定用户使用默认构造函数创建指定的对象，并添加到该场景中，让用户亲手实验中深入理解对象的真实含义，为以后面向对象思想的学习打下牢固的基础。该对象的属性不需要用户管理，先使用系统默认的属性。

内容：1.创建一个鱼的对象添加到大海里。

2.创建一个海龟的对象添加到大海里

3.创建一个鲸的对象添加到大海里

任务 2:

目的：规定用户修改已存在对象的某一属性，使屏幕上出现发生相应变化的源对象。让用户在即时变化中生动地体会到属性对于对象的重要性。

内容：1.将刚刚添加到大海中的鱼的坐标改成 (200,200) 假设该坐标与默认初始坐标有一定差别，观察鱼的位置有什么明显变化。

2.鲸鱼的颜色变成黑色的，看鲸鱼的外表将发生什么样的变化。

任务 3:

目的：对已存在对象调用指定的相应动作，用户通过调用已有函数实现任务要求的动作，比如跳，跑，游，飞，坐下之类的。使用户在自行调用函数的过程中理解函数的使用方式，并在看到函数调用后实现的效果，体会函数调用给对象带来的变化。此任务可以和任务

内容：1.对该鱼调用 leftMove(20)函数，使其能够向左移动 20px。并调用 upMove(40)，使其能够向上移动 30px。

2.让鱼在水里能够吐泡泡。调用 bubble () 函数

3.让鲸先浮到水面，再喷出水来。先对鲸这个对象调用 float()，再调用 pengshui()

任务 4:

目的：为某一特定对象添加和删除属性，比如为鱼增加一个 eyeColor 的属性，来达到可以改变鱼眼睛的颜色的目的。或者删除掉鱼的 color 属性，让我们无法改变再次访问到鱼的这个属性。让用户了解到对象的属性都是认为创建的，可以自行添加也可以自行删除，为以后继承的学习中在子类中添加特有属性做出准备。

内容：1.为鱼添加 eyeColor 这个属性，并初始化为 blue。

2.为龟创建一个 tailLength 的属性，让大家可以改变乌龟尾巴的长度。

任务 5:

目的：从一个已有对象继承出更多的对象，比如鱼可以有鲤鱼，草鱼，鲨鱼之类的，让用户在自行选择的过程中理解到面向对象思想的精髓内容继承，并为多态内容的学习打下坚实的基础。

内容：1.以鱼这个类为父类，创建出三种具体鱼类鲤鱼，草鱼，鲨鱼作为鱼的子类，他们享有父类的全部属性，并且可以为这三个子类添加相应的特有属性。并且将创建新建的子类对象添加到场景中。

任务 6：

目的：理解多态这一重要思想的含义，多态能够让对象在运行时决定实际调用的方法体。

内容：1.以龟为基类和其一个子类小龟，拥有一个同名方法 leftMove()，龟的 leftMove()默认每次动 30px，而小龟的 leftMove()默认每次动 10px，分别将基类和子类实例化并调用这个方法，显示的结果将随着方法所属类的不同而向左移动的间隔不同（假使二者在同一个 x_coordinate 上面）。

2.以鱼为基类，而任务 5 中创建的鲨鱼为子类，拥有同一个方法名字 bubble()，鱼的默认 bubble 的大小为 20px，而鲨鱼的 bubble 比较大是 50px，分别将基类和子类实例化并调用该方法，显示的结果将随着方法所属类的不同而 bubble 的大小不一样。

2.2.5 商店属性库 TABLE

记录商店属性中每个属性、函数、以及属性中的各个状态，对应的价钱。

3 游戏设计项目成员

林楠 11302010062 (组长)

杨君雯 11302010066

舒静 11302010063

