游戏设计文档

OOP 严肃游戏开发

11302010059 何文琦

11302010065 王治珍

11302010027 刘立

目录

1. 游戏概述

1.1 游戏简介

该款游戏为网页版 oop 教学拼装类游戏，采用 2D 金属风格画面设计，用户需要运用 oop 的知识闯过十关，才能完成最终任务。

1.2 游戏背景

故事发生在一个美好的 OOP 国度，骑士原本守护着可爱的 OOP 公主。突然某一天，OOP 国度被一群野蛮的怪兽入侵。它们不仅大肆破坏家园，甚至掳走了 OOP 公主。为了拯救被困的 oop 公主，骑士从废墟中重新出发，利用自己的智慧和勇气面对重重考验，竭尽全力去营救 oop 公主。营救途中需要经历十关的考验，每一关可能遇到的危险都是不同的，可能是有怪兽 boss，需要拼装武器才能打败 boss;可能是需要寻找零部件，组装打开一密室门的钥匙等等。在这个过程中，玩家必须对 OOP 的知识熟练掌握才能够过关，关卡设置会越来越难，最终玩家将要面对终极大 Boss，打败它才能最终拯救他心爱的 OOP 公主。

1.3 游戏特点

(1) 游戏不同于一般的严肃游戏，在游戏的过程中玩家还需要开动大脑去寻找画面中隐藏的线索，找到线索并且结合 oop 的相关知识才能顺利过关。游戏关卡设计巧妙有趣，充分调动玩家的积极性
(2) 游戏画面精美，场景逼真
(3) 玩家的操作简便易学，基本构成是简单地拖曳和点击。右键点击可以查看到隐藏物品的代码段和参数段，以便于更加深刻的理解对应的 OOP 概念。

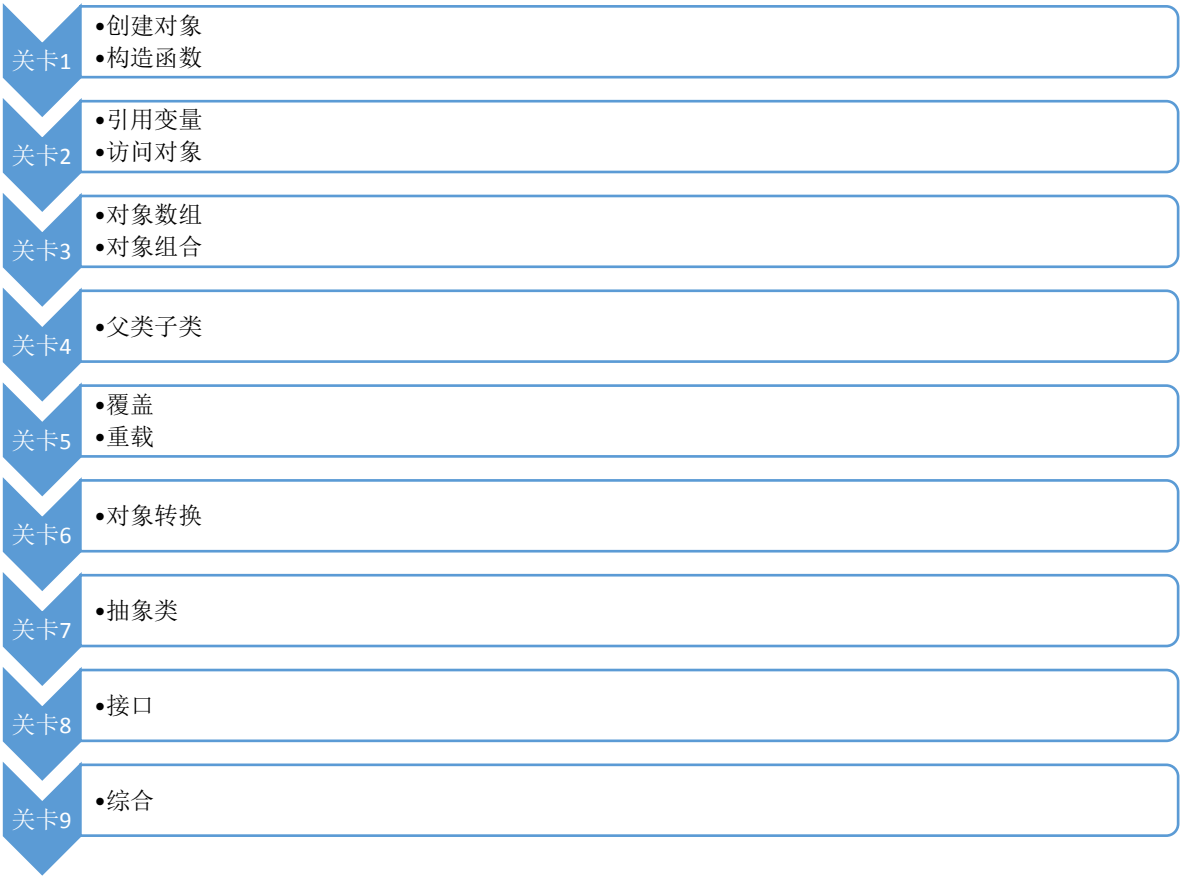
1.4 目标用户

有一定编程基础的计算机或软件专业学生

2. 游戏设计

2.1 oop 知识结构

知识结构图：



2.2 游戏关卡

2.21 第一关：翻腾的河流

旁白：我从废墟中醒来，之前发生的一切依旧还在脑海中。一群怪兽冲进了城堡，囚禁了 oop 公主，把我这个护卫扔了出来。我必须要重新回到城堡，拯救公主。城堡周围有条护城河，我得想个办法找到过河的工具。这时，有只老鼠窜了出来。

任务：用找到的馒头和老鼠交换一个充气筒。

旁白：环顾四周，看看还有什么可以用到的东西。

任务：在低洼处有一个神秘的箱子，用另外找到的绳子固定自己，下到低洼处。

旁白：箱子上有一些按钮，似乎可以制作什么东西。

任务：选择正确的语句，创建一个没有充气的充气船。沿着绳子爬回去，找到一把船桨。离开废墟来到护城河，用充气筒给船充气，乘船划过护城河。

```
class Boat{
    Boat(){
    }
}
```

2.22 第二关：难以翻越的城墙

旁白：在我面前的，是 oop 王国的城墙，大门紧紧地闭着。翻越它，我才能进到里面去。
任务：发现一块石碑，上面是灰，用在附近找到的鸡毛掸子将灰扫掉，发现上面写着 6C 61 64 64 65 72，破解这个密码，原来是要找到一把梯子。
旁白：6C 61 64 64 65 72？
任务：找到隐藏在城堡附近的梯子，设置它让它搭在城堡的墙上。
旁白：梯子搭好了，顺着它爬上去。
任务：设置梯子的承重，然后沿着梯子一步一步爬上去，中途一边爬可以一边增加梯子的高度。

```
class Ladder{
    int height;
    int weight；
    Ladder(){
        this.height=10;
        this.weight=0;
    }
    void addHeight(int a){
        height+=a;
    }
    void setWeight(int weight){
        this.weight=weight;
    }
void Putit(){}
}
```

2.23 第三关：颤栗的塔楼

旁白：我从城墙上落了下去，发现自己在一个塔楼面前，也许里面有什么线索。可是门口有一个护卫，看起来很厉害的样子。我想我需要利用附近的东西制作一件武器。
任务：找到散落在各处的武器图纸，拼成一张完整的魔法弓箭制作图。
旁白：塔楼的附近似乎有许多可以做武器的材料，也许之前发生过什么激烈的战斗。不管怎样，我需要根据图纸找到一些必须的材料。
任务：找到所需材料，一把弓臂，一根弓弦，一支箭，三团火合并的火球，按照正确的方式和数量把东西拼装起来，制作出了制作图上一模一样的魔法弓箭。将护卫打败，进入塔楼。

```
class MaggicWeapon{
    Bow bow;
    MaggicArrow maggicArrow;
    MaggicWeapon(Bow bow, MaggicArrow maggicArrow){
        this.bow=bow;
        this. maggicArrow=maggicArrow;
    }
    void Shoot(){}
}
```

```
class Bow{
    Limb limb;
    Bowstring bowstring;
    Bow(Limb limb, Bowstring bowstring){
        this.limb=limb;
        this.bowstring=bowstring;
    }
}
class MaggicArrow{
    Fire[] fire;
    Arrow arrow;
    MaggicArrow(Arrow arrow, Fire f1,Fire f2,Fire f3){
        this.arrow=arrow;
        this.fire=new Fire[3];
        this.file[0]=f1;
        this.file[1]=f2;
        this.file[2]=f3;
    }
}
```

2.24 第四关：塔楼内的密室

旁白：我走进房间，刚进入突然听到一声响，回头一看入口的门关上，再去试图打开已经不可能，我被困到密室中。密室中物品杂乱放置，寻找不到出口，我必须找到走出密室的方式。

任务：观察密室，四面已经都变成墙壁，房间中间有一个壁炉。此时看到一面墙上有一个闪闪发光的按钮，按一下，出现一个暗号输入框，提示可能需要找到暗号。

提示：右下角的提示中出现一个图，图上内容为一个保险箱

任务：需要找到这个保险箱，并且打开它，暗号可能藏在保险箱中，观察密室中的物品，在角落中发现目标保险箱。走向保险箱。

任务：在保险箱旁发现钥匙。拿起钥匙，试图打开保险箱。浮出提示框。

旁白：钥匙老旧。

任务：右击看到钥匙的参数代码，包括颜色，形状，材质，原材料需要从密室中寻找，密室中的物品，有一个破油漆桶，一个打磨工具，一块方形的铜，一块方形的铁，一块圆柱形的木头，要先选中和原钥匙相同材质的原材料即，方形的铜，再利用打磨工具打磨成圆环形，最后给它染上红色。

任务：拼装好新钥匙，打开保险箱，得到写着暗号的羊皮纸。回到发光的按钮旁边，输入暗号，壁炉里的火熄灭。发现把手，打开炉盖，发现楼梯，下楼过关

```
class Origin_key {
    String color;
    String shape;
    String material;
    Origin_key(){
      this.color = "blue";
      this.shape= "circle";
```

```
            this.material = "copper";
        }
        void turnLeft(){
        }
        void turnRight(){
        }
        void open(){
        }
        void print_out(){
        }
}
class Real_key extends Origin_key{
        Real_key(){
            super();
        }
        Real_key(String color,String shape, String material){
            this.color = color;
            this.shape = shape;
            this.material = material;
        }
        public void setColor(String color){
            this.color = color;
        }
        public void setShape(String shape){
            this.shape = shape;
        }
        public void setMaterial(String material){
            this.material = material;
        }
}
```

2.25 第五关：被侵占的武器库

旁白：沿着环形楼梯走到尽头有一扇门，我抬头一看门匾上写着“武器库”，门口还站着一个手拿武器的怪物守卫。

任务：打败守卫，进入武器库并从武器库中获得新式武器。

提示：可使用拥有以下技能：射枪（伤害值 1），发射激光（伤害值 2），炸弹（伤害值 3）

旁白：终于把守卫杀死了，打开大门走进去，选中武器准备离开，此时突然出现一个更加厉害的护卫

任务：杀死这个护卫才能离开武器库。需要学习新的技能（功夫，伤害值为 4）并且升级已有技能（射枪要升级到伤害值为 2，炸弹升级到伤害值为 3）

旁白：打败了怪物护卫，我离开武器库，接着向前走。

```
class Knight {
        int lifecnt = 100;//体力值；
```

```
        String name;
        Knight(){}
        Knight(String name){
           this.name    = name;
        }
        int Shoot(String target){
           return lifecnt-1;
        }
        int Bomb(String target){
           return lifecnt-2;
        }
        int Laser(String target){
           return lifecnt-3;
        }
        boolean win(){
            If(lifecnt==0)
                return true;
            else
                return false
        }
}
Knight_s extends Knight{
        public Knight_s(){
           super();
        }
        int Shoot(String target){
            return lifecnt-2;
        }
        int Bomb(String target){
            return lifecnt-3;
        }
        int Kongfu(String target){
            return lifecnt-4;
        }
}
```

2.26 第六关：塔楼里的宴会厅

旁白：路上经过许多仆人的尸体，形状惨烈。我继续前行，仿佛听到前面有哭声。跑向前去，发现有一个还活着的受伤的小女孩在哭泣，小女孩丧失语言能力。

任务：读懂小女孩手语，按照手语指示寻找药水。

旁白：宴会厅一片狼藉，在餐桌下面发现一个蓝色的药水瓶，瓶子上的标签上写着药的属性（救命还是毒药），药效指数（1，2，3），背面的标签还写着下一瓶药水的方位图。

任务：破解蓝色药水瓶后面的方位图，然后寻找下一瓶药水，方位图显示，下一瓶药水在三角形

结构里，需要找到三角形结构。

旁白：最左侧的餐桌上的刀叉是摆成三角形的，红色药水在那里，这又是一瓶毒药，后面还有一个图，看来还有第三瓶药水。

任务：破解示意图上只有的字谜：水火不相容，答案是藏在烛台里。

旁白：药水找齐了，但是蓝色和红色都是毒药必须转换为良药才能救命。

任务：查看药品的共同特征（均为同色同药效指数的药水），并且转换属性。房间内藏有魔法工具，该工具可以快速完成转换。但工具需要自己寻找。

旁白：三种药水混合后给小女孩服下，小女孩的伤痊愈。我想要从密闭的宴会厅出去却发现没有出口。这时小女孩拉着我走到一个餐桌旁，蹲下，默念咒语就打开了一条暗道，我从暗道出去。

```
class Medicine {
    String type;
    String name;
    int level;
    Medicine(){
        this.type = "good";
    }
    void Effect(){}
    void ChangeType(){}
}

class Poison_1 extends Medicine{
    Poison_1(){
        super();
        this.type = "bad";
        this.level = 1;
        this.name = "poison_1";
    }
}
class Poison_2 extends Medicine{
    Poison_2(){
        super();
        this.type = "bad";
        this.level = 2;
        this.name = "poison_2";
    }
}
class Poison_3 extends Medicine{
    Poison_3(){
        super();
        this.type = "bad";
        this.level = 3;
        this.name = "poison_3";
    }
}
```

```
Cure_1 extends Medicine{
    Cure_1(){
        super();
        this.type = "good";
        this.level = 1;
        this.name = "cure_1";
    }
}
Cure_2 extends Medicine{
    Cure_2(){
        super();
        this.type = "good";
        this.level = 2;
        this.name = "cure_2";
    }
}
Cure_3 extends Medicine{
    Cure_3(){
        super();
        this.type = "good";
        this.level = 3;
        this.name = "cure_3";
    }
}
```

2.27 第七关：帝王的后花园

旁边：我乘坐滑车滑了下去，周围一片漆黑。啪，我飞了出去，摔在了一片软软的地方，这好像是青草的味道。那这里大概是后花园了吧。我得想办法到宫殿里去。

任务：向前走，路灯下得到一把没有齿的钥匙柄。继续向前走到宫殿后门，发现上锁，得到的钥匙打不开。发现房梁上有个圆盘，高度够不着。需要回去拆下机械上的棍子，返回打落圆盘。

旁白：得到圆盘。总觉得这个色泽的机械哪里见到过。

任务：返回花园，某个边角有台小机器，中间空着一个圆盘。放入，出现谜题 1。解出谜题 1 得到提示。

旁白：蓝色憩息在忧伤之下，红色躲进了带刺的芬芳，绿色，绿色在哪里？光明指引着方向。

任务：去风信子(紫色)下取得蓝色钥匙，玫瑰中取得红色钥匙，路灯正对的方向取得绿色钥匙。装好正确的钥匙，打开门过关。

```
Class Lock{
    Boolean open(Key key){
        If(key.unlock() == 2)
            Return true;
        Else
            Return flase;
    }
```

```
}

Class Plate{
    Boolean down(Stick stick){
        If(tool.getHeight()>200)
            Return true;
        Return false;
    }
}

abstract Class Key{
    abstract protected int unlock();
}

Class RedKey extends Key{
    protected int unlock(){
        Return 1;
    }
}

Class GreenKey extends Key{
    public int unlock(){
        Return 2;
    }
}

Class BlueKey extends Key{
    protected int unlock(){
        Return 3;
    }
}
```

谜题 1 就用 OOP 练习题

2.28 第八关：微光的走廊

旁白：我蹑手蹑脚地打开了宫殿的后门，穿过一个房间，这是一条长长的走廊。走廊里灯影摇曳，好像随时就会熄灭一样（即本关限时）。

任务：反身去黑暗的转角，右侧是另一条路，得到一个工具箱，有钥匙孔。回到前面的走廊，一直向前，途中发现一扇废气的电门，以及三个损坏的发电机。再向前，找到一面墙，墙上内嵌着一个密码箱。点击箱子，弹出密码箱界面，需要四位密码，点击播放。

旁白：一二五六七八九。

输入：丢三落四，得到钥匙，打开工具箱。得到榔头、螺丝、扳手。反身，用榔头修理第一个发电机，扳手第二个，螺丝第三个。电门开启，过关。

```
interface Tool{
```

```
    abstract protected String fix();
}

class Hammer implements Tool{
    protected boolean fix(String damage){
        if(damage.equel(“铁板突出”))
            return true;
        return false;
    }
}

class Ascrew implements Tool{
    protected boolean fix(String damage){
        if(damage.equel(“拉闸掉了”))
            return true;
        return false;
    }
}

class Wrench implements Tool{
    protected boolean fix(String damage){
        if(damage.equel(“螺帽松了”))
            return true;
        return false;
    }
}

abstract class Generator{
    protected String damage;
    protected boolean isOK;
    public boolean generate(){
        if(this.damage.equel(“”))
            return true;
        return false;
    }
    public Boolean isOK(){
        return isOK;
    }
    abstract protected boolean getFixed();
}

class Generator1 extends Generator{
    Generator1(){
        damage = “铁板突出”;
        isOK = false;
```

```
    }
    protected boolean getFixed(Tool tool){
        if(damage.equel("") || tool.fix(this.damage)){
            this.damage = "";
            this.isOK = true;
            return true;
        }
        Return false;
}
}
class Generator2 extends Generator{
    Generator2(){
        damage = "拉闸掉了";
    }
    protected boolean getFixed(Tool tool){
        if(damage.equel("") || tool.fix(this.damage)){
            this.damage = "";
            this.isOK = true;
            return true;
        }
        Return false;
}
}
class Generator3 extends Generator{
    Generator3(){
        damage = "螺帽松了";
    }
    protected boolean getFixed(Tool tool){
        if(damage.equel("") || tool.fix(this.damage)){
            this.damage = "";
            this.isOK = true;
            return true;
        }
        Return false;
}
}

class Edoor{
    Generator[] generators = new Generator[3];
    Edoor(Generator generator1, Generator generator2, Generator generator3){
        generators[0] = generator1;
        generators[1] = generator2;
        generators[2] = generator3;
    }
    public boolean open(){
```

```
        for(int i = 0; i < 3; i++)
            if(!generators[i].isOK())
                return false;
        return true;
    }
}
```

2.29 第九关：闹剧的终焉

旁白：穿过废弃的密门，我来到了大殿的后方。大殿被洗劫之后空空荡荡，只有中央的座椅还留在那儿，而坐在中间的那个身影，我不会忘记！打败他，就能救出公主！他在睡觉。但是我依然小心，只有一次机会。

任务：不可走到椅子前方。在角落里得到一个螺丝刀，取得，发现地上有字。

旁白：成功往往是在最危险的地方。

任务：悄悄走到椅子后面，在椅子下得到一把枪。已经损坏。返回，发现柱子上有一块颜色不同，有枪印。放入枪，拉开，取出多种改装道具。改装，打死 BOSS。

提示：道具有多种组装方法，最后输出伤害能够打死 BOSS 即可。部分拼装产生一种新的枪的子类（即改装包），部分则为辅助接口。不同枪攻击不同，且需要不同拼装配合。又，某种枪，需要限时输入特定参数，连续多次增强攻击。

旁白：却见 BOSS 中枪后站了起来，转向我，我内心有一丝惊恐：难道他还活着？！他走向我，一步，两步，啪一声倒下了。我倒吸一口冷气。

```
class Gun{
    protected int Atk = 100;
    boolean attack(Aim aim){
        aim.setHP(aim.getHP() - atk);
        if(aim.getHP() <= 0)
            return true;
        return false;
    }
}

class RepeatingRifle extends Gun{
    boolean attack(Aim aim){
        while(randomQS())
            atk+=200;
        aim.setHP(aim.getHP - atk);
if(aim.getHP() <= 0)
            return true;
        return false;

    }
    boolean randomQS(){
        //随机传参的题目
    }
```

```
}

class LaserGun extends Gun{
	LaserGun(){
		atk = 1000;
}
	boolean attack(Aim aim){
strengthen();
		aim.setHP(aim.getHP() - atk);
		if(aim.getHP() <= 0)
			return true;
		return false;
	}
	void strengthen(){
		if(this instanceof Amplifier)
			atk *= 2;
	}
}
interface LaserAmplifier{}

class WaterGun extends Gun{
	WaterGun(){
		atk = 0;
}
	boolean attack(Aim aim){
strengthen();
		aim.setHP(aim.getHP() - atk);
		if(aim.getHP() <= 0)
			return true;
		return false;
	}
	void strengthen(){
		if(this instanceof water)
			atk = 10;
		else if(this instanceof acid)
			atk = 1024;
	}
}
class WaterConnon extends WaterGun{
	void strengthen(){
		if(this instanceof water)
			atk = 100;
		else if(this instanceof acid)
			atk = 2048;
	}
```

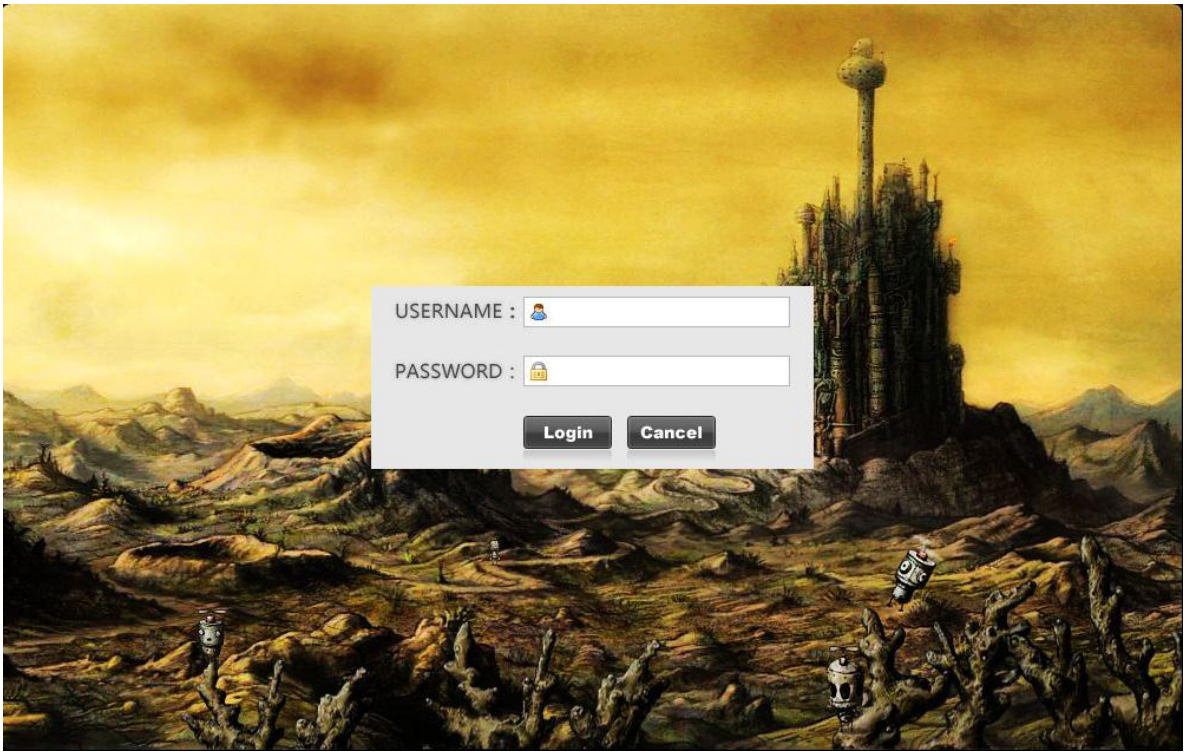
```
}
interface acid{}
interface water{}


interface Aim{
}
class Boss implements Aim{
	private int HP = 2000;
	public int getHp(){
		return this.HP;
	}
	public void setHP(int HP){
		this.HP = HP;
	}
}
```

2.3 游戏玩法

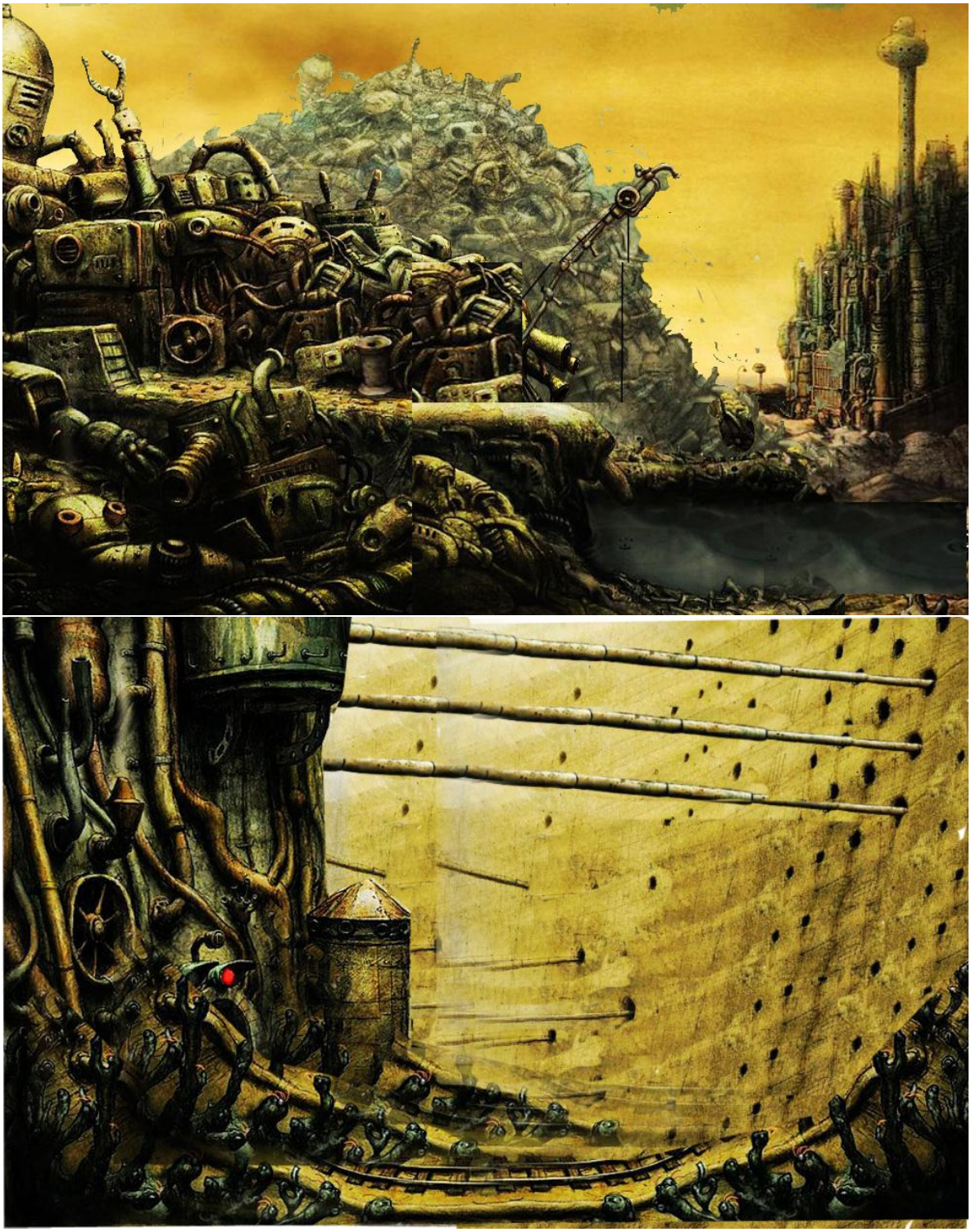
玩家在画面上寻找有效的物品，寻找过程中需要解谜等，找到物品后通过拖曳来拼接完成目标任务。完成一关才能进行下一关，该游戏提供游戏存档，保存玩家当前进度。

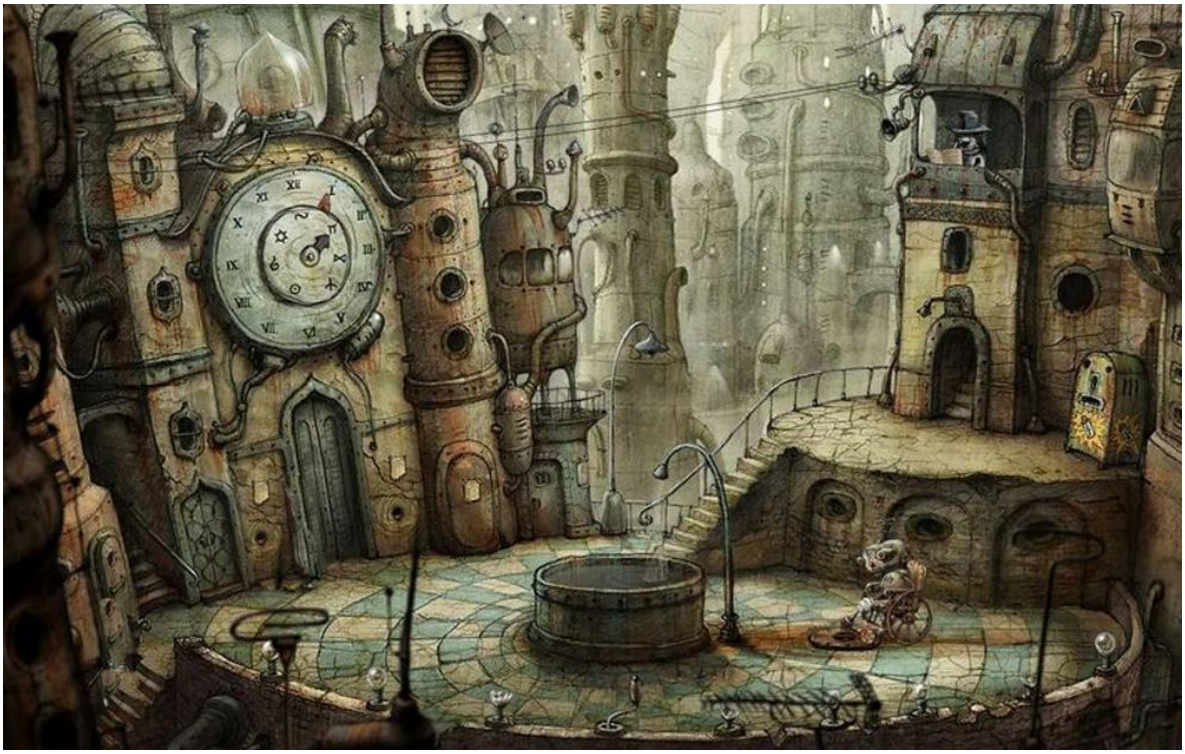
2.4 游戏界面

2.41 登陆界面



2.42 关卡界面






安卓网@HiAPK.com

（注：这些界面图只是暂定，之后会有调整）

3. 游戏后台

游戏逻辑的设计在上面的关卡设计中已经阐明，数据存储使用数据库。为每一个用户维护一个账号，一个密码和一个游戏进度。