目录

[Bunker Treasure Report 2](#_Toc28030534)

[游戏概述： 2](#_Toc28030535)

[游戏背景: 2](#_Toc28030536)

[游戏玩法： 2](#_Toc28030537)

[创新内容： 2](#_Toc28030538)

[开发工具： 2](#_Toc28030539)

[开发细节： 2](#_Toc28030540)

[美术任务细节 2](#_Toc28030541)

[程序任务细节 6](#_Toc28030542)

[项目拆分 7](#_Toc28030543)

[结果和思考： 9](#_Toc28030544)

# Bunker Treasure Report

## 游戏概述：

游戏的名字是《Bunker Treasure》，是一款2D像素解谜游戏，Cyberpunk和神秘风格。欲发布平台是PC、Android和IOS，第一版需要开发大约10关。

### 游戏背景:

故事发生了20XX年的Bunker基地，入侵者植入了许多智能机器人到Bunker基地，要偷取核心能量石，玩家既是主角，一个半机械半人类的守卫者，通过完成任务升级自己的技能，也可以收集能力碎片，抵挡外来入侵机器人。

### *游戏玩法：*

是玩家控制一个最多10X10格子场景的横竖变换。一方面阻止外来入侵者偷取Bunker中的Treasure，另一方面通过拼接和消除场景中的特殊格子来获得力量或者完成任务。获得的力量分为道具类和被动能力类。且道具又分为无目标使用类型，和指定目标使用类型。

## 创新内容：

1. 对Cyberpunk和远古神秘风格的像素绘制，以及像素动画的绘制表现
2. 对代码架构的创新，在Unity中尝试使用MVC的架构格式。
3. 对关卡的设计，以及入侵者的AI设计

## 开发工具：

开发引擎：Unity3D

开发语言：C#

开发IDE：Visual Studio 2017

像素绘制软件：Asprite

## 开发细节：

### 美术任务细节

作为一个美术风格为CyberPunk和神秘感觉的像素画来说，美术工作分为如下几个步骤：

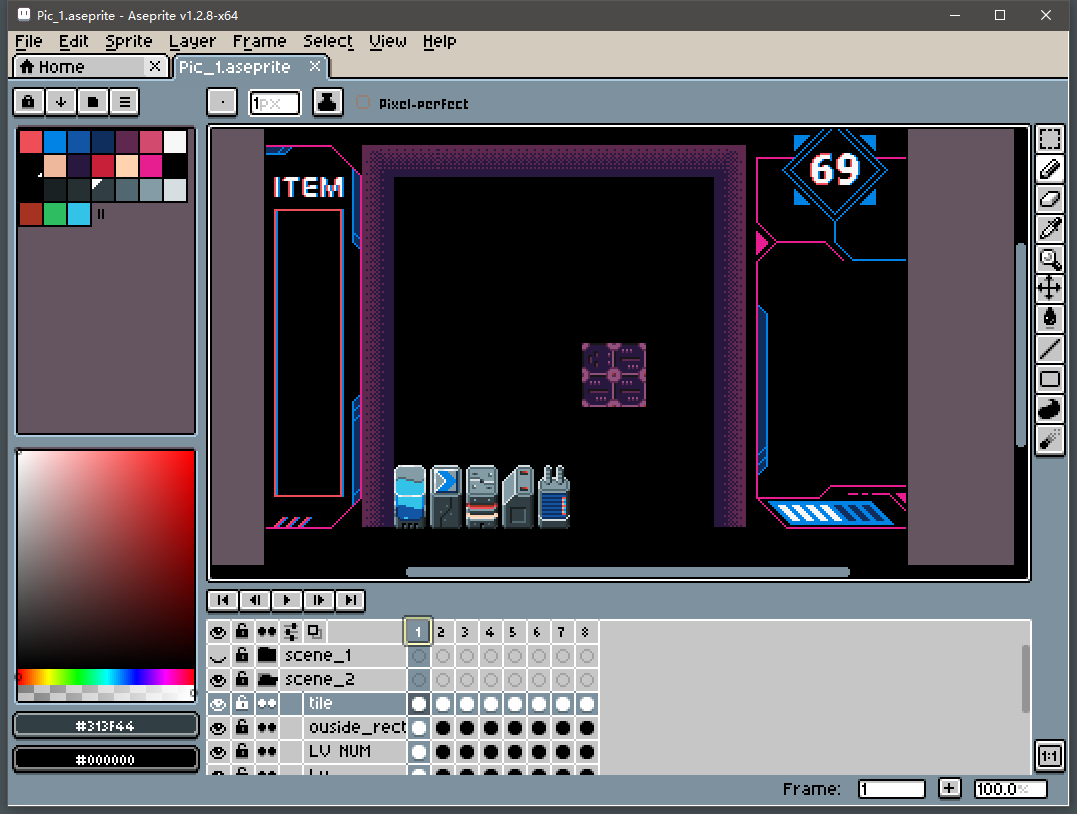
1. 首先寻找资源，确定感觉







1. 其次量化美术素材，进行绘制
2. 美术资源分层，输出序列帧

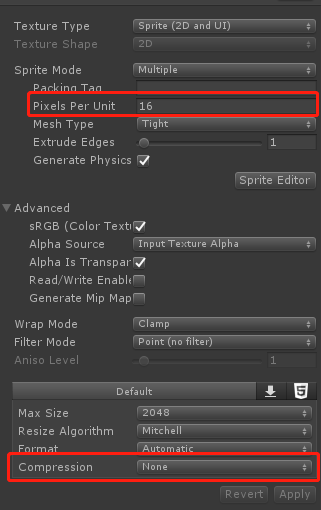
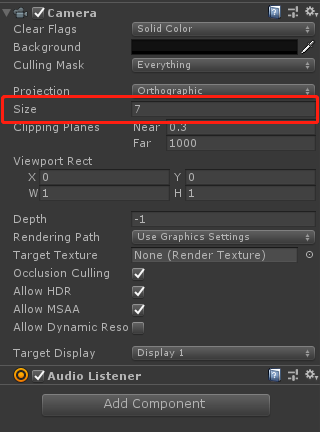




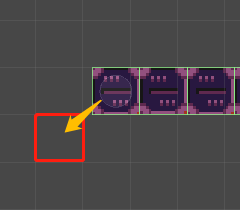
1. 整合到Unity3D中。

U3D中制作，需要注意的点：

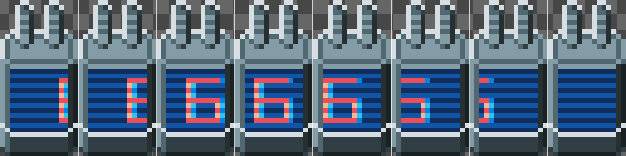
1. 首先在导入序列帧后，因为序列帧tile是按照16pixel去做的，所以应该 Pixels Pre Unit设置为16，这样，不会有像素误差，且适应各种分辨率。最后，可以调整Camera的Size，使Tile精灵图与美术概念图Layout相匹配。

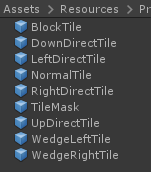
 

（PS：这个参数的意味是，如下图，在分辨率像素变化后，其实，Unity在屏幕中的单位是不变的，这样设置好对应关系后，可以将美术像素和游戏中单位化相匹配。如：默认是100，就是每个格子单位是100x100像素，如果tile为16像素，则在float时出现误差，难以在map中动态布局生成）



1. 上图中，还有一个就是Compression要设置为None，否则像素颜色会出现不正确的现象
2. 将序列帧导出到一张纹理上，方便切图制作序列帧动画，也可以加快渲染。



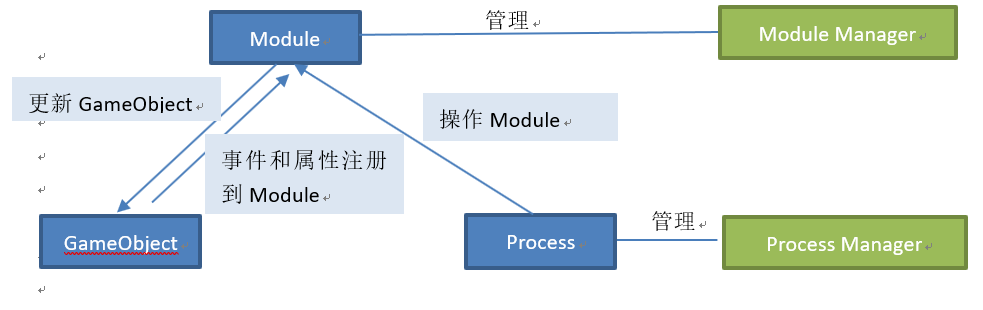
1. 可以用后处理的bloom增加画面的Cyber感表现
2. UI在制作时，Prefab中需要设置好Anchor，这样，在不同的设备，分辨率变化的情况下，可以保持UI相对的位置不会发生变化。
3. Tile和 UI的Prefab制作时，要与脚本名称相对应（因为实例化时用到了C#的反射机制）例如： 
4. 每一个Tile的动画的名称与相关的Tile的名称呼应。

### 程序任务细节

作为程序框架结构的创新尝试，本项目的程序是这样架构的。

#### 新框架概述：

框架分为3个模块：BasicProcess，BasicModule和Unity本身的GameObject。其中GameObject作为“View” 模块，BasicProcess作为“Controller”模块，而BasicModule作为“Module”模块，他们三者的关系是：



其中，作为“View”模块的GameObject，会将其节点和event注册到Module中，在Module中，进行逻辑层运算，更新GameObject的相关组件和属性。然后Process中调用工厂模式ModuleManager操作Module生成，销毁等，同时为了与Unity框架相对应，Process与Scene相呼应，通过Process来切换Scene，管理场景中逻辑的相关事务。ProcessManager用于Process的相互切换。

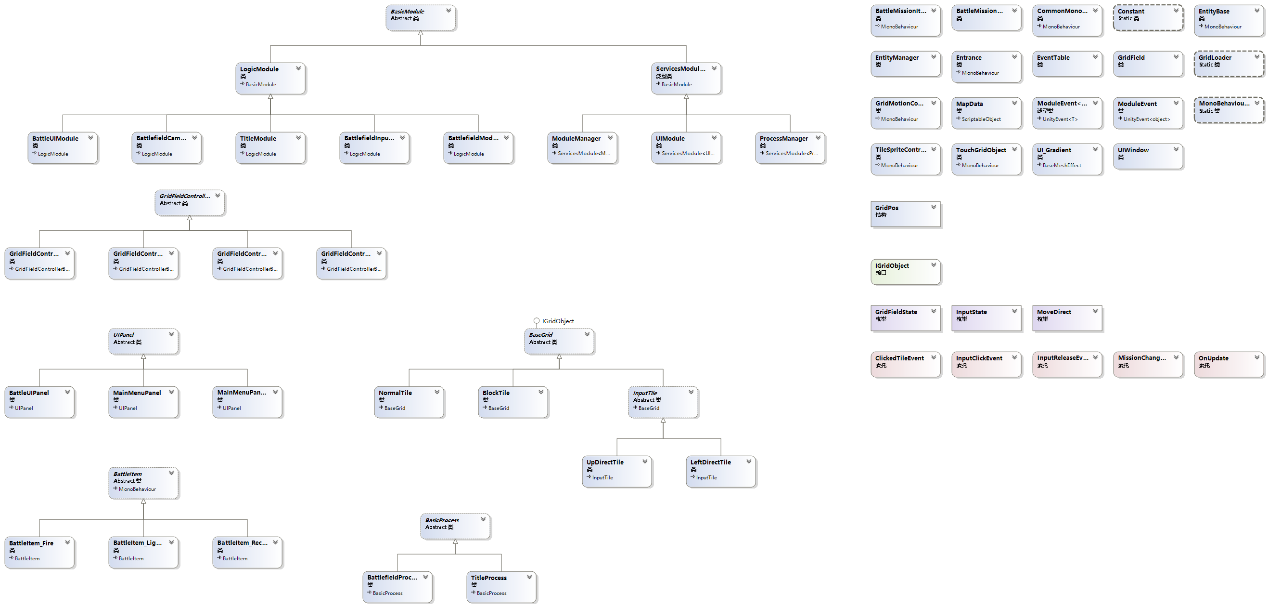
为了灵活使用Module模块，框架中又将Module派生出一个单件ServicesModule和一个LogicModule。Module通过C#反射机制，动态的Activator.CreateInstance出Module实例。

#### 新框架的优势：

传统的Unity3D开发中，程序员会根据Unity的GameObject和Component的设计方式。写很多继承自Monobehavior的各种功能的组件。在场景中，静态预先挂接到GameObject上，或者动态AddComponent到GameObject上。但是，这么做有一些弊端。对于协助完成游戏模块或者系统，在后期分析或者迭代项目时，无法理清多层级GameObject下挂接的众多组件之间的关系，也很难分析Component间相互的纠缠和耦合。所以，在Unity3D中尝试用MVC是一个很好的解决方案。

对于开发的程序员而言，清晰的结构更能显示出此框架的优势所在。在模块测试与调试阶段，由于逻辑间的低耦合，会使开发进度提升。而且在UI模块上，进行了高度封装，UIPanel和UIWindow，分别管理正常UI的GameObject和类似于模态对话框的UI。

类结构图如下：

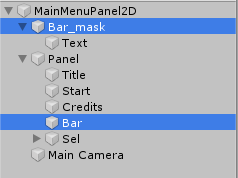


### 项目拆分

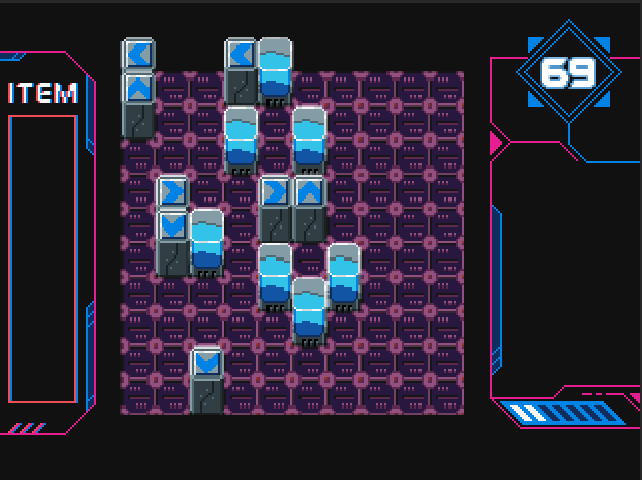
1. 项目为2D游戏项目，场景分为5个，Entance，LogoScene，MainMenu，MainGame，EndGame
2. Entance是入口场景，通过Entance.cs脚本来控制Manageres启动和初始化所有模块和Process，通过不同的Process来切换场景。
3. LogoScene是展示我们StudioLogo的过渡场景
4. MainMenu是主菜单页面，主要是UGUI交互和表现逻辑



通过UGUI的Mask和原图叠加制成条纹状滚动文字。



1. MainGame是游戏是主界面，效果图如下，分为如下几个部分：



1>左侧道具栏，支持拖拽滚动。

2>右侧关卡任务栏，右下为关卡进度条，右上为当前关卡序号。

3>中心为游戏操作界面，网格中特殊物品点击后支持横竖条带移动，且特殊物品相互靠近后可以双消或者三消。消除后给玩家提供力量。场景网格中会出现机器人，机器人会以一定的能力靠近场景中的能量宝石，玩家通过控制网格条带的移动，阻止机器人偷取能量宝石。

4>背景会有一些点缀用的特效或动画。

1. EndGame是游戏的结束界面，在当前关卡游戏完成之后（胜利或者失败），游戏会进入到EndGame的场景中，通过EndGame场景，玩家选择“Replay”“Next”还是“Quit”。

如果之后要进行商业化，可以在Replay和Next选项点击时，增加广告AD的播放，为开发人员提供一些微薄的收入。

游戏会在PlayerPrefs中记录游戏当前的关卡数和其他数据，在玩家再次登录游戏时可以读取。

## 结果和思考：

这个项目是一个新的尝试，包括新的构思和新的框架架构。在游戏玩法方面，不但融入的新的益智和消除元素，还在谜题中，加入了一个不确定性的盗窃能量的机器人，使游戏更有新颖的感觉。通过：移动（操作输入，变换谜面）、消除（变换谜面和获取能力）、完成任务（关卡目的驱动）、获取释放能力（给予玩家选择权和玩家本身成长因子）、阻碍敌人（游戏压力）众多游戏元素构成一个完整的游戏有机体。

游戏美术方面是一个复古的像素风格，符合独立游戏和小工作室的美术工作量。加上一些Cyberpunk化的感觉，可以表现与主题贴近的背景。且，像素2D赛博朋克感觉一直也是本人想做的一个风格。

游戏程序方面，采用了新的框架架构，利用C#的特性，和U3D的传统的开发流程上的创新，对于小规模，单机，后期迭代开发，离散开发有很强的优势。也在调试Bug中可以更好的定位Debug。

思考：其实这个游戏更像是一个实验品Demo，因为这个游戏的故事内容相对较少，情节和情感的相关内容也几乎没有，关卡是设计和益智成分是主要内容。对于游戏而言，这仅仅是用于一个游戏玩法，在可靠粘性的玩法中玩家可以获得乐趣，但是，如果有一些故事背景再，反而，可以直击玩家内心的情感，让玩家感悟游戏的弧光。也许那样的游戏才会更加完整，沉浸感更高。