

Graffiti War

BY 林松青 周春阳

A decorative background featuring a variety of colored circles in shades of blue, red, yellow, green, and purple, scattered across the left side of the slide. A stylized rocket ship, composed of a blue body and a red nose cone, is positioned in the center-left area, pointing towards the right.

1

创作原因

我们想制作一款有别于飞机大战，跑酷等传统的控制台游戏。于是我们从一款手机游戏的游戏形式中获得灵感，开始创作Graffiti War。

Graffiti War 完全基于控制台开发且不使用任何贴图，但是其精美的UI，多样的特效与动画不同于人们印象中传统的控制台程序，让人耳目一新。其操作简单，但是可玩度高，同时富有趣味。

Graffiti War



点击鼠标左键以开始游戏

游戏中按空格以暂停游戏

双缓冲技术

避免在帧率过高的情况下出现闪屏的情况



与数理知识结合的核心算法

减小直线运动误差算法
根据动点计算角度算法
碰撞检测

搭建简单2D游戏框架



GSprite: 实现对于图像的各种操作
GAnimationSprite: 实现了动画的播放
GDraw: 实现图像资源管理和图像绘制
GMenu: 实现了功能完备的菜单



绘图技术

仅使用Windows api
绘图函数绘制了精美的UI界面及动画特效

简单2D游戏框架核心功能介绍

GSprite:

1. 使图像沿某一方向以某速率移动
2. 使图像绕某一热点旋转
3. 渲染图像
4. 缩放图像

GAnimationSprite:

1. 实现了动画的逐帧播放
2. 实现了多样化的动画模式

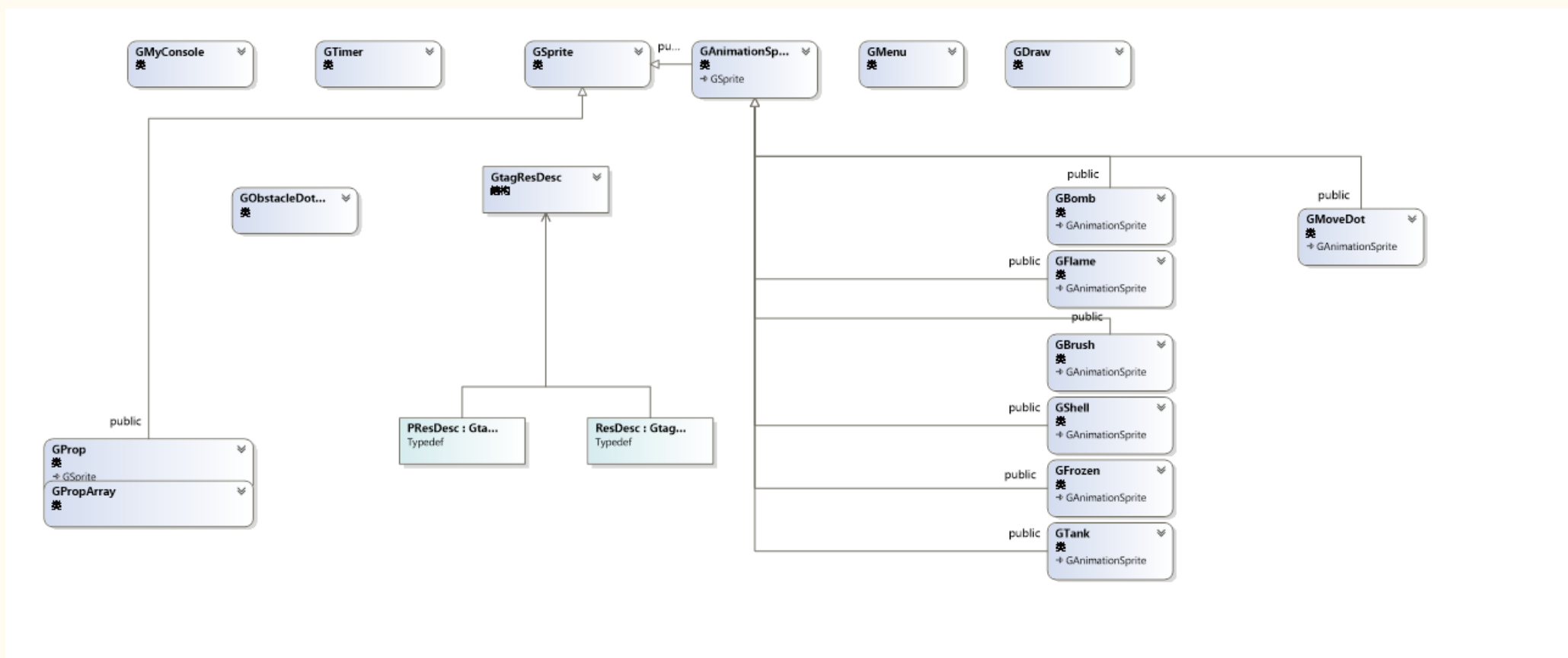
GDraw:

1. 统一存储和分配图像资源
2. 统一绘制所有程序需要的图像

GMenu:

实现了功能完备的菜单
如开始游戏，退出游戏，暂停游戏，以及重新开始游戏等功能

类图展示：程序一共含有18个类



与数理知识结合的核心算法的难点攻克

直线运动算法：由于直接使用计算直线长度公式产生较大误差，会导致飞机移动不协调，因此借鉴了Bresenham算法，调整直线运动

根据动点的角度计算：利用三角函数诱导公式，解决了窗口坐标系与标准坐标系坐标轴位置不统一的问题。然后利用解析几何的知识，计算出两点之间的角度关系。同时，将判断条件细致化，解决了两点距离较近图像抖动的问题。

碰撞检测：点与线段的碰撞中，利用多个矩形划分窗口区域，以达到确定线段在窗口位置的目的，精准确定碰撞范围

1. 六种趣味道具，形式多样，效果各异，每个道具效果独立进行而不会阻塞程序。



2. 交互界面全部可视化，即使是控制台，操作依然简单优雅，UI界面也同样精美。



开始菜单

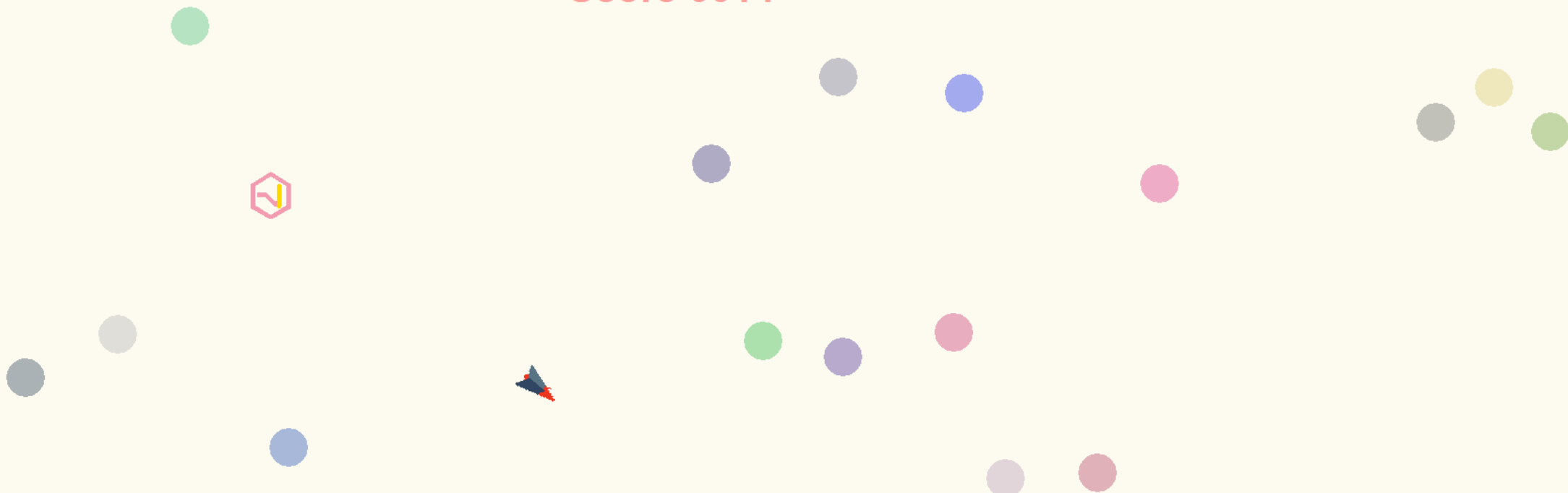


结束菜单

3. 功能设计人性化，增加游戏趣味性

- 1) . 随着游戏进程，障碍物点的生成会逐渐增多
- 2) . 不同道具的出现概率不一样，局内道具越多，新道具生成概率越低
- 3) . 游戏内有计分系统，让游戏更具挑战

Score 0014





THANKS