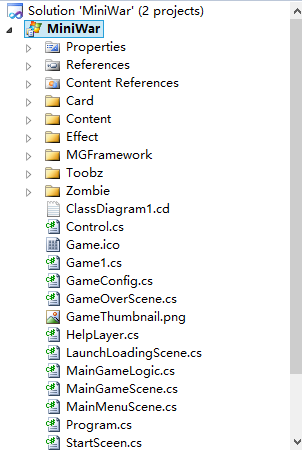
# 僵尸来了 说明文档

## 程序概述

该游戏一共有5个关卡，每个关卡多有不同数量及难度的僵尸靠近袭击你的房子，在任意关卡里都可以连接电线，购买和使用道具。整体游戏参考植物大战僵尸类似的场景以及接水管的玩法。用xna4.0编写，在vs2010环境下运行。游戏通过点击鼠标来更改电线朝向，如果左右两边的电线连通则改条线路通电如果僵尸在该路线上则对僵尸带来伤害，你需要防御住僵尸让他们远离你的房子。

# 整体构架

## 目录结构



Card文件夹里面是所有和技能卡片相关的代码

Effect 文件夹里面是闪电技能效果代码

MGFramework 文件夹里面是游戏的底层代码Minigame游戏引擎

Toobz 文件夹里面是接水管部分的代码

Zombie 文件夹里面是僵尸部分代码

为放在文件夹里面的代码是游戏逻辑场景等代码

## 游戏引擎介绍

MiniGame是基于Cocos2d-x for xna（cocos2d-xna）的思想进行开发的，只实现了部分简单功能。

游戏的入口为Game1.cs

在Game1中初始化和游戏引擎相关内容及鼠标显示等内容

public class Game1 : Game

{

GraphicsDeviceManager graphics;

SpriteBatch spriteBatch;

public Game1() : base()

{

graphics = new GraphicsDeviceManager(this);

Content.RootDirectory = "Content";

this.IsMouseVisible = true;

//这些

MGDirector.SharedDirector().Graphics = this.graphics;

MGDirector.SharedDirector().Content = base.Content;

MGDirector.SharedDirector().Landscape = true;

MGDirector.SharedDirector().DisplayFPS = false;

}

protected override void Initialize()

{

base.Initialize();

}

public void CompleteLoading()

{

MGDirector.SharedDirector().RunWithScene(new TestingScene());

}

protected override void LoadContent()

{

spriteBatch = new SpriteBatch(GraphicsDevice);

//这些

MGDirector.SharedDirector().Initialize();

CompleteLoading();

}

protected override void UnloadContent()

{

}

protected override void Update(GameTime gameTime)

{

if (Keyboard.GetState().IsKeyDown(Keys.Escape))

Exit();

//这些

MGDirector.SharedDirector().Update((float)gameTime.ElapsedGameTime.TotalSeconds);

base.Update(gameTime);

}

protected override void Draw(GameTime gameTime)

{

GraphicsDevice.Clear(Color.CornflowerBlue);

//这些

MGDirector.SharedDirector().Draw();

base.Draw(gameTime);

}

}

游戏概念与游戏引擎中类的对应关系

|  |  |
| --- | --- |
| 概念 | 引擎中对应的类 |
| 导演 | MGDirector |
| 摄像机 | MCamera |
| 场景 | MGScene |
| 布景 | MGLayer |
| 角色 | MGSprite |
| 动作 | MGAction |

导演

对应的是MGDirector，在整个游戏里只有一个导演游戏开始和结束时都需要调用MGDirector的方法完成游戏的初始化或者销毁工作。MGDirector提供了一些管理场景的方法，如果RunWithScene，ReplaceScene等，MGDirector还有另一项功能就是获取一些系统信息设置，如得到屏幕大小，Content等。

摄像机

队形的类为MGCamera。MGCamera在游戏中比较重要，每一个MGNode都需要使用MGCamera。当节点发生缩放，旋转和位置变化等时候，都需要MGCamera参与。有了摄像机节点才会被选软成大家能看的到的，例如背景和人物角色等。

场景

场景在minigame里对应的类为MGScene。MGScene中存放的是需要渲染的布景，人物角色和彩电，它使其做为一个整体一起渲染一起销毁一起被场景切换使用。

MGScene的类结构如下图

MGNode

MGScene

MGScene类本身只有初始化和生成的代码没有其他，它在游戏中主要只是承担的是一个容器的功能。游戏开发中把多个需要绘制的对象放在MGScene里进行统一管理。MGScene继承与MGNode。

MGNode的主要函数如下

Virtual void AddChild(MGNode child);//增加节点

Virtual void RemoveChild(MGNode child);//删除节点

MGAction RunAction();运行动作

Void StopAction();停止动作

因为MGScene继承与MGNode，所以MGScene拥有了MGNode的一些方法，这样就可以通过AddChild方法把渲染的对象添加到MGScene里面了。

布景

布景在游戏中对应的类为MGLayer，它是一个层的概念每个游戏中都可以有很多的层，没一层负责各自的任务游戏中有专门放道具，放电线，响应鼠标的层。

MGNode

MGLayer

MGLayer类的主要函数解析如下：

Virtual void TouchBegan();//鼠标按下

Virtual void TouchMove();//鼠标移动

Virtual void TouchEnd();//鼠标抬起。

用MGLayer创制出丰富的交互性界面。

人物角色

人物角色在游戏引擎里面对应的类为MGSprite，精灵精灵是整个游戏开发处理的主要对象，地上跑的僵尸，通电用的电线，道具卡片，卡片爆炸效果等等等这一起的一切都是精灵，精灵可以是一个可以不断变化的图片。这些变化包括位置的变化旋转大小缩小和运动等。

MGNode

MGSpire

MGSprite类的主要函数解释如下

Virtual void Draw()；//绘制

Static MGSprite SpriteWidthTexture();//根据纹理生成图片

Static MGSprite SpriteWithSpriteFrame();//根据Plist生成文件

有了精灵的产于游戏就将不再是静止的了，而是充满生机的，有各种僵尸的袭来，充满智慧的电线，和各有威力的道具

动作

动作在游戏里对应的类为MGAction，僵尸从左边移动到右边就是MGAction的一个子类MGMoveTo，在游戏中MGAction就是节点的行为表现。

MGAction类的主要成员解释：

bool isEnd//动作是否完成了

static MGAction Action();//生成MGAction

virtual void Stop();//停止动作

## 主游戏实现

### Card: 游戏道具

##### Paper.cs 类是道具的父类

Paper 的成员如下：

MGSprite Sprite; //卡片精灵

IntVector Vector; //Grid位置坐标

Virtual void Lauch(float dt); //检测

Virtual void Detonate(); //释放效果

Virtual void SetVector(int x,int y); //设置Grid位置坐标

Paper和他的子类

Paper

BigBombPaper

BigIcePaper

BombPaper

IcePaper

RandomWirePaper

RocketPaper

XWirePaper

##### CardShowLayer.cs

class CardShowLayer : MGLayer

{

Void Update(float dt);

//……………..More

}

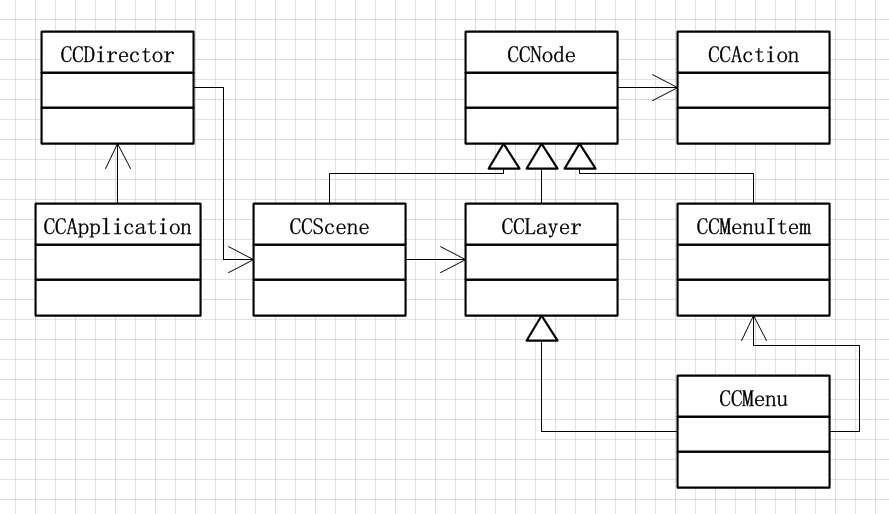
CardShowLayer作为道具卡片的容器会被添加到MainGameScene中，它完成了道具卡片的呈现，在update中调用每个Paper的Lauch方法循环更新检测看有没有需要出发的卡片技能。

### MGFramework

CCNode是Cocos2d-XNA中最重要的对象。CCNode的addChild()方法，我个人认为是最重要的方法。它充当了一个容器的角色可以addChild进去无数个节点，但每个节点只有一个父节点，同时他可以执行回调函数和CCAction有postion,scale,rotation等等属性。

CCLayer，它是CCNode的子类,层是一个可以绘制的区域.可以将ui,sprite等对象添加到其中。

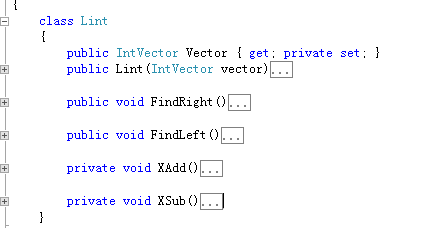
CCMenu从CClayer派生而来,可以处理输入消息,CCMenu包含多个CCMenuItem对象,在接收到输入消息之后,根据触摸位置,将输入消息分发到相应MenuItem,执行回调函数，CCMenu还提供一个对齐菜单项的功能



CCSprite， 可以看成是某一时刻,一个2D的图片的一个剪辑部分

### Toobz

##### Lint.cs



电闸线头类， Vector表示线头的位置线头分为左右线头，在游戏中是不显示的只作为游戏判断。FindRight左边线头出发看有没有右边连同线头，FindLeft反向。

##### Wire.cs

class Wire

{

UpdataImage();

Drop(int x);

ChangeWireDirection();

ChangeWireState(WireType type);

}

游戏中会密密麻麻的在场景里铺满电线我们需要拿这些电线制造电流

UpdataImage的作用是更新图片显示，每当点击电线的时候都会让电线构成顺时针旋转变化。

Drop每当有新的电线产生是需要给他一个正确合理的位置

ChangeWireDirection 让电线旋转触发他的同时会触发UpdataImage改变图像显示

ChangeWireState 在使用部分道具后电线会发生变化，变化成type形态的电线

##### WireShowLayer.cs

class WireShowLayer

{

bool TouchesBegan();

void WireIsChange();

//………………………..More…

}

TouchesBegan在点击发生时判断每个Wire找到点中的哪一个让他发生旋转,在发生旋转后WireIsChange会触发判断是否有练成功的电线如果有产生闪电。

### Zombie

##### Actor.cs

class Actor : MGNode

{

MGAnimate StandAni;

MGAnimate RunAni;

MGAnimate DieAni;

void Run()

void Stop()

virtual void Dead()

void ChangeHp();

}

StandAni站立动画，RunAni行走动画，DieAni死亡动画，

Run();行走 Stop();停止 Dead();死亡 ChangeHp() 僵尸会在你的花园里上游走你需要干掉他，如果让他们触电他们就会掉血，失血过多的时候就会播放死亡动画完成死亡Dead();

Actor

ActorBabe

ActorBoss

ActorFast

ActorLr

ActorLv

ActorNormal

ActorSb

##### ZombieShowLayer.cs

class ZombieShowLayer : MGLayer

{

void Thunder(object wireShow);

//……………..More……………….

}

WireShowLayer层的电线连同后会掉用Thunder方法检测是否有僵尸在电线攻击范围内如果有让改僵尸产生伤害。

## 总结

主游戏是通过多个MGLayer分别是WireShowLayer, ZombieShowLayer, CardShowLayer构架起来的结构清晰，每个MGLayer完成相对独立的模块，将其加入到MainGameScene中完成整个游戏。