# 游戏介绍

赛道

障碍

道具

角色

# 场景切换

# 设计思路

## 表结构

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NodeName | id | efflist | filename | width | height | windex | hindex | movementud | movementlr |
| Block | 0 | 10 | cloud.png | 1 | 1 | 0 | 0 | 0,true,1,50 | 1,false,0,0 |
| Block | 1 | 11 | building\_1.png | 2 | 2 | 0 | 0 | 0,true,1,50 | 1,false,0,0 |
| Block | 2 | 12,17 | puff\_1.png | 1 | 1 | 0 | 0 | 0,true,1,50 | 1,false,0,0 |
| Block | 3 | 10 | player\_float.png | 1 | 1 | 0 | 0 | 0,true,1,50 | 1,true,0,2 |
| Block | 4 | 10 | player\_float.png | 1 | 1 | 0 | 0 | 0,true,1,50 | 1,true,1,2 |
| Block | 5 | 10 | puff\_2.png | 1 | 1 | 0 | 0 | 0,true,1,50 | 1,true,0,2 |
| Block | 6 | 11 | building\_4.png | 1 | 1 | 0 | 0 | 0,true,1,60 | 1,false,0,0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NodeName | id | effobj | effproperty | value | effecttime |
| Effect | 10 | 0 | 0 | -80 | 0 |
| Effect | 11 | 0 | 0 | -120 | 0 |
| Effect | 12 | 0 | 0 | -40 | 0 |
| Effect | 13 | 0 | 0 | 100 | 0 |
| Effect | 14 | 0 | 1 | 100 | 0 |
| Effect | 15 | 1 | 2 | 50 | 1500 |
| Effect | 16 | 0 | 3 | 100 | 3000 |
| Effect | 17 | 0 | 4 | 0 | 100 |

## 数据结构

typedef enum {

player = 0,

other = 1

} EffectObject;

typedef enum {

hp = 0,

score = 1,

speed = 2,

distance = 3,

movement = 4

} EffectProperty;

typedef struct {

int type;//上下，左右，

bool enable;//不可移动

int direction;

float speed;

} Movement;

typedef enum {

NormalBlock = 0,

BigBlock = 1,

DizzyBlock = 2,

LeftBlock = 3,

RightBlock = 4,

TrackBlock = 5,

HighSpeedBlock = 6

} BlockType;

typedef enum{

Coin,

Invincible,

HP,

Magnet

} ItemType;

typedef enum {

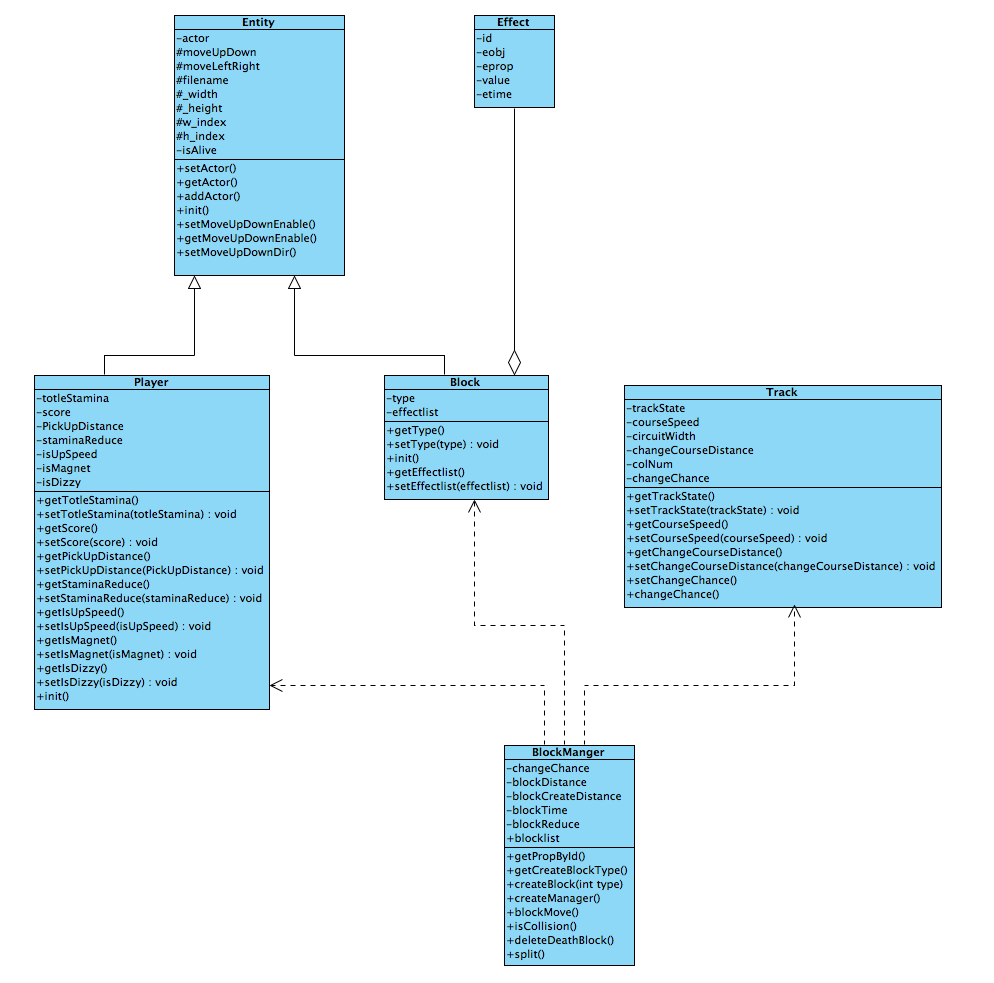
Left = 0,

Middle = 1,

Right = 2

} TrackState;

## 类的设计与实现



模块设计

## 实现技术点（关键点）

生成障碍的位置保证不重叠 角色有路可走

道具生成，不能与障碍距离太近，道具的生成机制与障碍的不同

变道时场景中节点的显示位置

追踪障碍的实现（单独创建一个管理列表，每帧检测一次角色运动方向，设置追踪障碍的运动方向）

障碍生成之后带有移动（左，右，追踪），有可能跟其他障碍或道具重合，高速障碍的重叠

无敌或加速状态的实现，体力不减，其他所有物体速度增加

碰撞检测的精准度

## 程序流程



## 缺点与不足

面向对象的思维有欠缺，对类的属性和方法的理解，对数据封装的理解，对抽象统一的理解

对程序的可扩展性

对配表的思维，数据更改对程序重写 重编 重构的影响

思维不系统，想到一点补一点，思维不全面

数据和功能分离，散落的数据变量，感觉比较乱

# 重构

表结构

数据结构

typedef enum

{

SPEED\_X, //X方向速度

SPEED\_Y, //Y方向速度

STAMINA,//体力

SOCRE,//分数

COLLISION\_SIZE\_X,//X方向碰撞距离

COLLISION\_SIZE\_Y,//Y方向碰撞距离

MAGNET\_SIZE\_X,//X方向吸收距离

MAGNET\_SIZE\_Y,//Y方向吸收距离

}AttributeType;

typedef enum

{

WAITING, //等待生效

EFFECTING, //正在生效

OVERDUE, //已经过期失效

}EffectStatus;

typedef enum

{

ADD,

MUL,

EQUAL,

Lock,

}EffectNumType;

类结构

