基于 Unity 的手机横版跳跃游戏开发

吴一飞 郭晓丹

四川大学锦城学院 计算机与软件学院 四川 成都 611731

【摘 要】：本文设计了一个基于 unity 游戏引擎与 C#脚本语言开发的手机横版跳跃游戏，本游戏开发的过程大概可以拆分成十三个部分，包括：场景与瓦片地图的制作，设置碰撞体积与 2D 刚体参数，角色行动等脚本编写，设置视角与相机追踪，添加地图光源与光线渲染算法，设计角色动画树，添加可收集要素与障碍物，编写粒子特效与相机震动，设计游戏输赢目标与行为反馈，游戏 UI 制作，添加音效，以及最后的封装和移动端适配。本文中所用模型等素材均出自 unity 官方免费素材。

【关键词】：unity；C#脚本；着色器编程；unity 渲染；手机游戏

引言

近几年随着互联网技术的蓬勃发展，游戏行业也在逐步扩大用户规模与市场份额。而开源免费的 Unity 游戏引擎是一个对开发者来说非常友好的平台，在其上制作游戏也能够充分的参考海内外其他开发者的开发经验。并且使用 Unity 制作的游戏能够在各个平台上流畅运行。

因此本文基于 Unity 开发了一个平台跳跃游戏，并且最终可以在个人电脑与安卓手机上完整运行。在游戏开发中涉及到了 Unity 组件、C#脚本编写、光线渲染、着色器编程等方面的知识与内容。

1 具体开发与算法设计

1.1 关卡设计与人物设计

本文将要阐述制作的是一个横版过关类的平台动作游戏。确定了游戏的基本玩法之后，要进行的工作就是关卡设计。在 unity 中内置了一个相当方便的工具，即 tile map。通过该工具可以将提前放置在资源文件夹中的地图资源在网格视窗中进行可视化编辑。

新建一个平台图层之后将它设置为静态以确保该平台不会进行移动，同时在右侧检查栏中将图层优先级设置为最高。之后便可以打开引擎内置的调色板窗口，将资源中的地板贴图拖进调色板绘制场景的大概框架。值得注意的是，为了避免地图过于简洁不美观，本文使用了平铺地图随机化 API 来让地板贴图能够进行随机变化。算法思路为创建一个数组通过函数产生随机数，返还地图位置信息与旋转信息。

横版跳跃动作游戏中很重要的一环就是确保人物能站在平台之上，因此就需要用到组件平铺地图对撞机，它的作用是围绕每单个地图块建立对撞效果，当然，也要将该组件设置为静态，否则已有的瓦片地图将会因物理引擎作用而掉落。

此时刚刚放置入场景的人物素材只是一个精灵，同上节操作一样，首先将人物的图层切换为第二优先级的玩家角色图层。之后再将他变为一个物理实体，让他能够进行如跳跃攀爬等物理反馈。

向人物添加一个组件 2D 刚体后，为了使平台与人物的进行非常精准的碰撞，在组件之中设置碰撞检测为连续检测。同时为了方便插值，将内部板的值设置为在允许的地方进行插值。

此时人物已经具备了物理特性，如重力、摩擦力等，但没有设置具体参数，所以需要使用组件 2D 碰撞盒来为他添加相应的值。为了防止玩家角色因为摩擦力卡在墙体之间，将碰撞盒中的材质参数设置为铁氟龙。同时，为避免玩家角色在 Z 轴旋转，设置一个 Z 轴方向的约束。

想让人物移动起来，需要编写移动脚本。在脚本内部，首先将所有的输入方式如手机、游戏手柄包装在一个类中，该类负责提取所有不同类型的输入并且创建变量如水平移动和判断按键。同时该脚本将会检查建立的每一个环境框架和对玩家周围环境的影响。自动地对角色左脚进行两个射线投射，对蹲伏状态做一个射线投射。使用 character controller 移动角色，播放行走动画，（当然此时还没有编辑角色行走动画）。

值得一提的是墙壁攀爬动作，需要三个投射的复合体，预先计算最小爬升点积，当射线垂直向下击墙时，脚本将判断角色能够到达墙的顶部。

总的来说，运动分为两类，第一类运动为地面运动，分为站立运动与蹲伏运动；第二类为跳跃运动，分为普通跳跃、蓄力跳跃和攀爬。脚本会首先获取玩家的所有输入然后在执行，从而让代码更加流畅。

1.2 锁定视角与光照变化

此时角色已经可以进行操作了，但是玩家视角无法追踪他的移动。对于横版游戏来说，相机应该始终保持在比摄像头区域或平截头体区域大的水平，因此需要利用一个名为 Cinema Sheen 的工具来设置相机追踪。更改投影从正交变为透视图。此时视角就拥有了即时视差滚动，设置跟踪人物精灵便可以设为锁定视角。

Unity 中光源分为方向光、点光源、聚光源和面光源。能为场景提供光照效果。本次使用的为点光源模拟火把与壁炉的照明效果。

因为前向渲染虽然能获得更好的光照体验，但是会呈指数级的增加场景的绘制次数，因此难以用于移动设备，所以本文将项目设置为延迟渲染。

1.3 动画树

此时玩家角色虽然可以进行运动，但不会做出相应的动画。创建一个精灵表，其中包含适合不同动作的不同动画同时让所有框架的规格地图和它们保持同步。再使用蒙皮网格物以及实际游戏中用于角色生成平面的网格渲染器，进行诸如网格变形之类的操作。拆分网格，将角色的手、躯干、头部分开。再将角色模型与角色精灵合为一个对象，单击该对象可以分出单独的骨骼数据。

网格渲染器上已经具有拥有法线贴图的材质，因此可以通过网格变形来让骨骼弯曲并且移动从而达到制作动画的目的。

设置好前置步骤之后，接下来进行播放动画脚本的编写。在脚本中首先需要引用此前设置好的如刚体、移动等组件，定义不同动作的动画 ID 和它们的哈希等效项。同时设置不同的运动降低速度或增加速度。定义一个播放音频的方法，如当角色调用跳跃动画时播放对应的音频。

Unity 有一个强大的组件名为 Blend Tree 即混合树，能不同程度的综合所有动画的各个过程来混合动画，如人物在往右跑时突然往左跑。在混合树中创建事件只需要点击按钮然后在检查器中选择脚本执行的调用方法。

因为在动画之间混合，所以每个动画只是单个关键帧，但是在普通动画之间可以使用多个关键帧动画在它们之间融合，用混合树进行弥补空隙。

1.4 游戏机制

此时游戏已经有了大致的框架，但仍然欠缺机制的内容，如陷阱、收集品、通关门等。因此需要导入一些具有不同行为的游戏对象，如会击杀玩家的尖刺和可以收集积分球的神龛。同样，导入预先设置好的预制件，将它们拖拽到场景上。

在本文工程中，游戏通关需要收集放置在不同位置的三个神龛中的球，从而打开通关的大门。

想要让游戏机制生效，需要用到新的脚本。首先添加游戏失败，判断当玩家碰到陷阱或者坠落时实例化死亡效果，将玩家游戏对象设置为 false，随后播放游戏失败音效。脚本中多次进行碰撞检查，为了避免字符串比较，特别是在移动设备上，将陷阱作为整形储存在内部。

此时当角色触碰到陷阱时将会死亡，但无法重新开始闯关，因此需要某种方式来让游戏重新开始。

在脚本中设置一个类，由它储存玩家尝试挑战次数、收集情况。当玩家触碰陷阱时重新加载该关卡。该类持有自己的参考，并且有一个私有的静态变量，不会共享其他将要使用的变量。

2 后期处理与封装

2.1 后期效果

为了增加项目的一些视觉效果，现在需要进行后期处理。在组件中添加一个后处理层，为了获得图形和视觉效果在其中添加对撞机。在渲染后期处理量中可以自由新建配置文件随意添加想要的效果。

再添加一个脉冲监视器组件，即当脉冲源放置在场景某处时侦听的相机。在 Unity 中，脉冲监视器可以视作耳朵，音频源从场景的某处播放音频然后接收该音频。本文中想要达到的效果是在收集和玩家角色死亡的时候摇晃相机进行动作反馈。

所有的收集物上都有对撞机，当检测到玩家收集了积分之后会生成爆炸粒子并且进行动画处理的脚本。首先编译一个协程函数，定义一个对象获得震动的固定点坐标，返回 XYZ 轴的值，根据每一帧协程计算将震动数值返回到 camera 上。

接下来要做的工作是为游戏添加混音。游戏开发是编程、设计、音频三位一体的工作。现在需要编写一个音频管理器脚本。

将 AudioClip 数据封存在属性中，通过 bool 变量来判断此时需要播放的音效。该音频管理器将在运行时生成自己的音频源，它将设置环境片段并且进行设置，如播放随机的足迹音频。如此只需要几个音频源就可以覆盖项目中几乎所有音频，而不是拥数百个音源。

在游戏中，有一个 UI 可以让玩家知道目前失去了多少生

命值或者游玩了多少时间或者此时还有多少个收集物需要收集。Unity 中可以用 UI 编辑器直接进行编辑。在层级浏览器中创建一个画布，画布具有渲染模式和将屏幕空间整个覆盖的组件。在绘制时将创造可以放置在场景中的游戏对象。将 UI 锚点放置在希望放置的地方，简单的调整大小，使宽高比调整到适合移动设备的比例。同时建立一个新的 UI 场景脚本，引用一个 animator 参数抓取器，抓取散列的字符串以提高效率。

2.2 封装和移动端适配

在进行移动端适配时，需要设置一左一右两个可触摸操控按钮。同样通过 C#脚本来达到目的。运用由 MonoBehaviur 提供的方法检测对按钮的点击触发事件，钳位数据并且在移动图像的同时进行归一化范围应用轴向盲区。

此时游戏已经完成，可以进行封装了。点击 file 中的 build 按钮，在窗口中选择 android。ADK 将自动识别并且构建一个可以运行的 apk。

总结

本文对基于 unity 的游戏开发过程和问题进行了详细阐述。同时将过往的游戏经验带入到本次开发过程中，让玩家能够获得良好的游戏体验。

参考文献：

[1] 任建邦.基于 Unity3D 的手机游戏客户端的设计与实现[D].北京交通大学,2013

[2] 王树斌.浅析 Unity3d 开发游戏流程及常用技术[J].电脑知识与技术,2012,008(022):5351-5352