

物理大作业——动能势能的极限转化

灵感来源 & 实现功能：

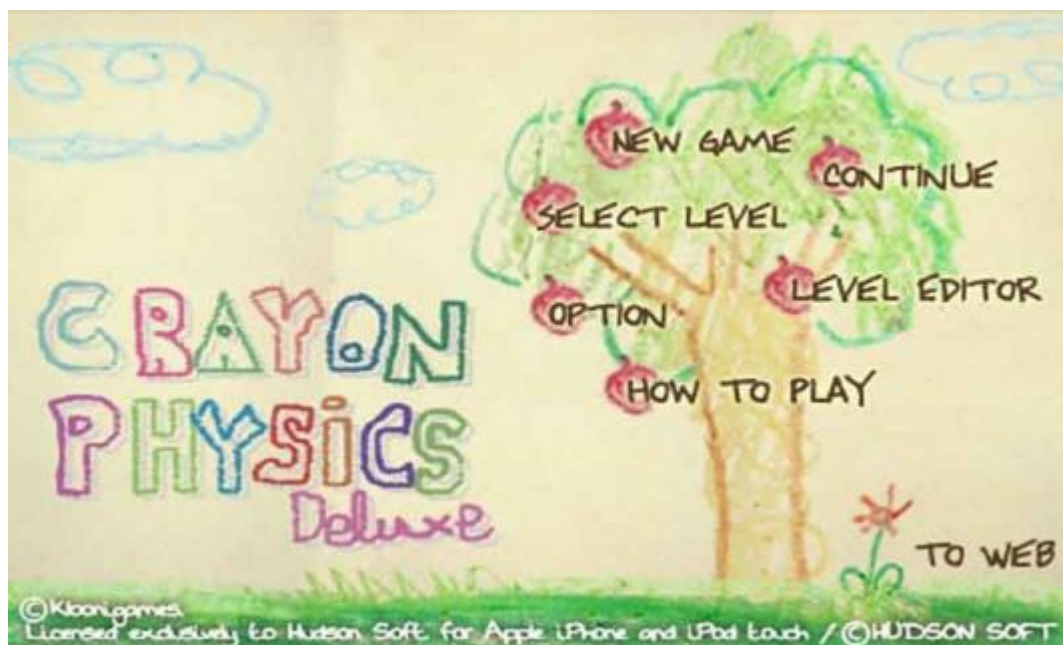
大作业的灵感来源于日常益智类小游戏，从我们自身特长考虑，将物理原理和计算机技术相结合，并且将原先的游戏在一定程度上加以改进，让玩家能够有一些不同的游戏体验。在进行游戏的同时，可以让玩家在一定程度上对一些物理规律（如：动能势能之间的转化、引力场的运用、刚体碰撞等）有所认识和探究。

大作业类型：实用体验类，模拟物理场景的数值模拟类型。

游戏原型：

蜡笔物理：这是一个神奇而有趣的游戏！幻想有一天你拥有一支神奇的魔法彩铅笔，你想用它做做什么呢？玩过这个游戏你会感到很多事情不可思议哦！

（同学试玩体验：通过简单的操作，你可以运用你的智慧，通过一个个不违背物理规律的方式将小球送到终点处，有兴趣的话可以尝试一下智商被凌辱的快感）



愤怒的小鸟（太空版）：在太空版游戏中，这些小鸟们将会换上全新的太空装备，而且游戏的玩法会有所改变，虽然还是用弹弓，但是因为身处太空，

重力环境与地球完全不同，万有引力会对游戏造成很大的影响。

（同学试玩体验：这是他们团队对于蜡笔物理这款游戏的改进部分的原型，通过加入一些引力源或是引力场，来达到一些意想不到的效果）



小组成员介绍及任务分配：

| 物理内核 | 图形化处理 | 文档&展示 | 关卡设计 |
|------|-------|-------|---------|
| 谢天成 | 徐世超 | 周宇皓 | 谢天成、蒲逸飞 |
| 蒲逸飞 | 高必成 | | 徐世超、高必成 |
| | | | 薛震东、蔡万鑫 |
| | | | 刘志健 |

所涉及的物理学现象&规律：

- 1、刚体碰撞
- 2、势能动能转换
- 3、万有引力
- 4、斥力表现（类磁力）

所使用工具介绍：

最终我们的物理引擎采用的是Box2d；图形库采用的是OpenGL和SDL。

Box2d:

Box2D是一个用于模拟2D刚体物体的C++引擎。Box2D 是一个用于游戏的 2D 刚体仿真库。从游戏的视角来看，物理引擎就是一个程序性动画(procedural animation)的系统，而不是由动画师去移动你的物体。

核心概念：

- 刚体(rigid body)

一块十分坚硬的物质，它上面的任何两点之间的距离都是完全不变的。

- 形状(shape)

一块严格依附于物体(body)的 2D 碰撞几何结构(collision geometry)。形状具有摩擦(friction)和恢复(restitution)的材料性质。

- 约束(constraint)

一个约束(constraint)就是消除物体自由度的物理连接。在 2D 中，一个物体有 3 个自由度。如果我们把一个物体钉在墙上(像摆锤那样)，那我们就把它约束到了墙上。这样，此物体就只能绕着这个钉子旋转，所以这个约束消除了它 2 个自由度。

- 接触约束(contact constraint)

一个防止刚体穿透，以及用于模拟摩擦(friction)和恢复(restitution)的特殊约束。你永远都不必创建一个接触约束，它们会自动被 Box2D 创建。

- 关节(joint)

它是一种用于把两个或多个物体固定到一起的约束。Box2D 支持的关节类型有：旋转，棱柱，距离等等。关节可以支持限制(limits)和马达(motors)。

- 关节限制(joint limit)

一个关节限制(joint limit)限定了一个关节的运动范围。例如人类的胳膊肘只能做某一范围角度的运动。

- 关节马达(joint motor)

一个关节马达能依照关节的自由度来驱动所连接的物体。例如，你可以使用一个马达来驱动一个肘的旋转。

- 世界(world)

一个物理世界就是物体，形状和约束相互作用的集合。Box2D 支持创建多个世界，但这通常是不必要的。)

OpenGL:

OpenGL (全写Open Graphics Library) 是个定义了一个跨编程语言、跨平台的编程接口规格的专业的图形程序接口。它用于三维图像（二维的亦可），是一个功能强大，调用方便的底层图形库。

OpenGL作为当前主流的图形API之一，它在一些场合具有比DirectX更优越的特性：



- 与C语言紧密结合：

OpenGL命令最初就是用C语言函数来进行描述的，对于学习过C语言的人来讲，OpenGL是容易理解和学习的。如果你曾经接触过TC的graphics.h，你会发现，使用OpenGL作图甚至比TC更加简单。

- 强大的可移植性。

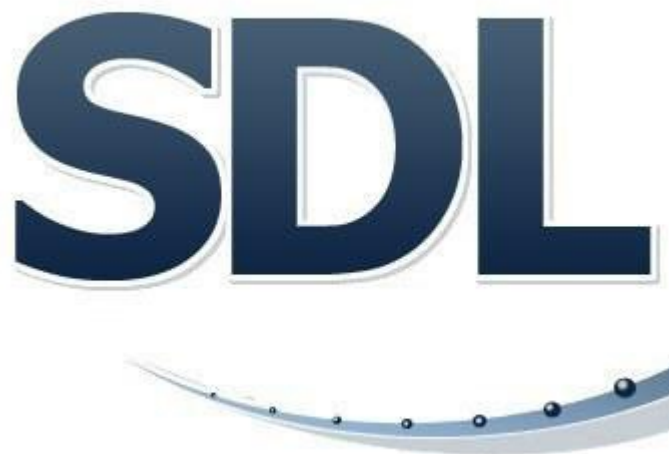
微软的Direct3D虽然也是十分优秀的图形API，但它只用于Windows系统（现在还要加上一个XBOX游戏机）。而OpenGL不仅用于 Windows，还可以用于Unix/Linux等其它系统，它甚至在大型计算机、各种专业计算机（如：医疗用显示设备）上都有应用。并且，OpenGL 的基本命令都做到了硬件无关，甚至是平台无关。

- 高性能的图形渲染。

OpenGL是一个工业标准，它的技术紧跟时代，现今各个显卡厂家无一不对OpenGL提供强力支持，激烈的竞争中使得OpenGL性能一直领先。

SDL:

SDL (Simple DirectMedia Layer) 是一套开放源代码的跨平台多媒体开发库，使用C语言写成。SDL提供了数种控制图像、声音、输出入的函数，让开发者只要用相同或是相似的代码就可以开发出跨多个平台（Linux、Windows、Mac OS X等）的应用软件。目前SDL多用于开发游戏、模拟器、媒体播放器等多媒体应用领域。



另外我们在这次大作业中尝试的软件：windows GUI、QT、OPENCV、ACLLIB、Direct X.

制作过程中产生的问题：

时间紧迫：

这是我们遇到的问题中最为主要的部分，之后一切的问题都可以说是时间带来的影响。考试结束到大作业deadline之间时间差较小，并且我们小组中有些成员是ACM竞赛的队员，在此之间有参加校队的训练，让我们在代码实现上有些许困扰，但是我们依然在较短的时间内，将我们的应用程序设计达到一个可行的程度，而且这个工作是有一定的代码量和我们需要不停学习的地方。另外，时间问题也导致了我们的应用没有之前我们计划的那么精美，所以还请体验者多多包涵。

代码实现（学习问题、各自实现和拼装的问题）：

在基于时间不够的基础上，导致了各种问题自然而然的产生了。其中比较重要的是代码的问题，毕竟我们是用C++来实现这个游戏的物理引擎，即使是对于计算机竞赛的老队员来说，这和往常的竞赛方式不同，也是带来一种不同的体验和学习过程中的艰辛。另外在实现图形化的过程中，我们第二组的队员自主学习了许多和图形库有关的东西，在否定了之前考虑的软件过后，我们的队员义无反顾的继续学习其他软件的一些应用方式，才有了我们后来的工作成果。最后还要提一些关于拼装的问题，我们各组实现的是不同的部分，在完成各组自己的debug后，我们在拼装的过程中也遇到过各式各样的问题，我们正是在相互查错中才完成最后的成品。

关卡设计的难度：

由于ACM班的特殊招生过程，我们中并不是所有人都是对计算机有着非常深入的理解的，就像小编这样，但是我们同样在这个过程中发挥着我们的作用，除了写这个document以外，我们亲自体验了蜡笔物理这个游戏来源，力求能够完成符合要求的关卡设计。最初我以为这是零基础同学的福利，可在实际体验后才发现，其实并不是这样的。蜡笔物理这个游戏本身，因为游戏玩家的操作复杂多变，所以它需要一个合理的容错率，每一个复杂关卡的设计都需要通过精确地分析和计算，同时在策划的时候通过的引力场的加入更是给这个游戏加入了更多的不确定性，也正是在这样的情况下，才会强行提升游戏的设计难度。

未来改进方面：

界面设计（贴图及纹理）：

由于时间的关系，我们只能初步将可视化界面完成。在之后的过程中，我们会在界面设计方面多做改进，优化游戏界面，加入美工等一些有意义的改进，提高用户的游戏体验。

环境物理量参数调整：

对于我们所有团队成员来说，这是初次进行游戏设计。开始阶段，在我们并没有考虑游戏设计下，不知道环境物理量如何设计才能达到较为良好的游戏体验。这是要通过一些精密地数值计算完成的较为复杂的任务，因为时间原因无法在考试结束的短时间内完成。我们在之后游戏设计的优化中，对于环境物理参量的调整也是必不可少的一方面。

关卡难易度调整&难度阶梯优化：

同样因为时间原因，我们只完成了三关游戏设计以及制作。在未来的优化中，必然会有对关卡的补充，以及难易度的调整，才能更加有实用性和推广性，同时我们还会着眼于对于bug的修复，对于一些违反物理规律现象的修正，将关卡设计的更为合理，有难度梯度，降低或提升游戏的容错率，并且提出一些更具创意性的改进，引入更多的物理规律。

附录1（参考资料）：

关于游戏灵感：

引用百度百科词条：蜡笔物理、愤怒的小鸟太空版

关于物理引擎：

引用百度百科词条：Box2d

Box2d学习：<http://ohcoder.com/blog/categories/box2d-tutorials/>

Box2d Wikipedia：<https://en.wikipedia.org/wiki/Box2D>

相关网站：<http://box2d.com>

关于图形化处理：

引用百度百科词条：OpenGL、SDL

SDL学习：<http://tjummyk.github.io/sdl-tutorial-cn/index.html>

OpenGL官网：<http://www.opengl.com>

附录2（开源代码）：

附上Github地址超链接 —> [Physics Project](#)