

I. 实现功能：

A. 模拟篮球、地板、墙壁、篮板、篮圈的物理引擎（支持一定程度的加减速模拟）。

B. 3ds 模型的导入（用在导入篮球架和球网）

C. 多种视角（观众视角、球员视角、沙盘视角）和对应的控制模式。

D. 预设的投篮动画。

E. 背景音乐和音效（篮球与板（地板、墙壁、篮板）的碰撞声、篮球与篮圈的碰撞声、球员视角的脚步声、进球的刷框声）。

F. 地板的镜面效果,篮板的镜面和半透明效果,。

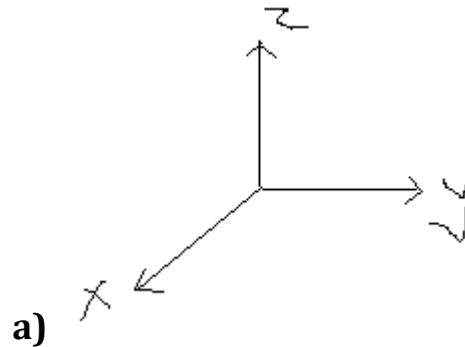
G. 聚光灯和阴影

H. 观众席和观众

II. 实现细节：

A. 物理引擎：

1. 坐标系：



如图，与 3dsmax 的坐标系、本程序绘图时的坐标系都相同。

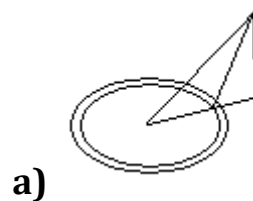
2. 包含的物体类型：

a) 动态的球（篮球）

b) 静态的法方向为坐标轴的板（地板、篮板、墙壁）

c) 静态的 xy 平面上的圆环（篮圈）

3. 圆环与球的碰撞检测：



如图解三角形就可以了

4. 物理引擎的步进:

a) 步进间隔为 0.001 秒, 意味着模拟真实时间每秒需要调用引擎的 **advance** 函数 1000 次, 对于 40 帧/秒的本程序而言, 模拟真实时间时每次重绘将伴随着 25 次引擎的 **advance**, 若是调用 1-24 次, 则意味着不同程度的减速, 若是调用次数达到 26 次, 则意味着不同程度的加速, 本程序允许这样的加减速, 每次重绘时引擎的调用次数属于 $\{x | x \geq 0 \ \& \ x \leq 50 \ \& \ x = 5 * a, a \text{ 为整数}\}$ 。

5. 篮球的旋转

- a) 篮球保存一个自己的旋转矩阵。
- b) 绘图时使用 **glMultMatrix***(本程序中使用的 **float** 型的矩阵, 所以是 **glMultMatrixf**) 与原矩阵相乘后绘图。
- c) 需要旋转时先载入单位矩阵, 接着使用 **glRotate***和 **glMultMatrix***对原矩阵进行旋转, 最后通过 **glGetFloatv** 取出 **OpenGL** 计算出的新的旋转矩阵。

B. 3ds 模型：

1. 使用网上现成的篮球架模型，转换成 3ds 模型，减少了大量多余的顶点，拆分为篮架模型（**basket.3ds**）和篮网篮圈模型（**net.3ds**）。
2. 使用网上现成的代码导入。
3. 球入框时，球网会根据球速有适当的偏移动画。



C. 视角和控制方式:

1. 视角:

a) 观众视角:

(1) 从观众席上看向篮球，篮球运动时，视线将强制追踪篮球。

(2) 可通过鼠标滚轮调节观察距离（相当于望远镜）。

(3) 当观察距离较小（程序中为小于 5 米），篮球将始终位于镜头中心（紧紧跟随）；当观察距离较大（程序中为大于等于 5 米），镜头望向篮圈高度的一半（减少了镜头的上下移动，方便纵观全局）。

b) 球员视角:

(1) 从球员的角度自由观察，无须追踪篮球，类似于 **fps** 游戏的效果（更合适的说法是类似于 **fps** 游戏死亡后自由观察的效果--）。

(2) 在此模式下可以进行自定角度、力度的投篮。

c) 沙盘视角：

(1) 从天上俯视篮球，篮球运动时，视线将强制追踪篮球。（暂时其实没什么实际用处，一开始是想着如果是真正的篮球的游戏，可能有人希望从这一类似沙盘的视角观察战术）

(2) 可通过鼠标滚轮调节观察距离。

2. 控制方式：

a) 通用控制：

(1) -键减速

(2) =键加速

(3) Tab 键切换视角，顺序为观众视角→球员视角→沙盘视角→观众视角……

b) 观众视角：

(1) W、S、A、D 与上、下、左、右分别使篮球在 $-x$ 、 x 、 $-y$ 、 y 方向的速度增加。

(2) 空格键使篮球在 z 方向的速度增加。

c) 球员视角：

(1) W、S 与上下进行前后平移。

(2) A、D 进行左右平移。

(3) 左、右进行左右旋转。

(4) E、Q 改变视线初始位置高度。

(5) Page Up、Page Down 改变视线仰角。

(6) 按住鼠标进行拖动也可进行转动。

(7) 按住鼠标左键可调节力量，放开鼠标左键时进行投篮（因为可选的力量不连续，远投极难，建议近距离打板）。

d) 沙盘视角：

(1) 从天上俯视篮球，篮球运动时，视线将强制追踪篮球。（暂时其实没什么实际用处，一开始是想着如果是真正的篮球的游戏，可能会有人希望从这一视角观察战术）

(2) 可通过鼠标滚轮调节观察距离。

(3) 左、右进行左右旋转。

D. 预设投篮动画：

1. `、1、2、3、4、5、6、7、8、9、0 为预设的罚球线投篮，力量由轻至重。

2. u、i、o、p 为预设的侧面打板。

E. 背景音乐和音效：

1. 背景音乐为 Kanye West 的 Amazing，循环播放，音量不变，用在这里感觉比较带感儿。

2. 篮球与板的碰撞声：

a) 篮球碰到地板、墙壁、篮板反弹时会发出碰撞碰撞声，音量大小与碰撞时垂直平面的速度和碰撞位置到观察者的距离有关。

3. 篮球与篮圈的碰撞声：

a) 篮球碰到篮圈时会发出碰撞声，音量大小与碰撞时的球速和碰撞位置到观察者的距离有关。

4. 球员视角的脚步声：

a) 球员视角时，如果高度合适（程序中为 1.2 米至 2.5 米）并且在 xy 平面上移动，会发出脚步声。

5. 进球的刷框声：

a) 进球时会发出刷框声，音量与球速和碰撞位置到观察者的距离有关，同速度、同距离下，空心进球的音量是非空心的两倍。

6. 对篮球的声音使用了左右声道的立体声效果。

F. 地板的镜面效果,篮板的镜面和半透明效果：

1. 由于尝试创建的环境是 **Staples Center**，而实际上那里的地板有反光效果，所以构建了地板的反光。



实际图片



程序效果图

2. 篮球场上真正有反光能力的其实不只是地板，还有篮板，程序中实现了篮板的镜面效果和半透明效果。



实际图片



程序效果图

3. 球场上有两个篮架，其篮板皆可实现这个效果，考虑到效率因素，默认只有 y 轴正方向处（观众视角右侧）篮板有反光和半透明，否则可能会比较卡。

4. 镜面效果的雾化

a) 原本使用蒙板来实现镜面的效果，无法区分远近。现在使用雾化的方式来使离镜面较远的镜像变模糊。



近距离效果图



远距离效果图

5. [键开关 y 轴负方向处（观众视角左侧）篮板的反光。

6.]键开关 y 轴正方向处（观众视角右侧）篮板的反光。

7. \键开关地板的反光。

G. 聚光灯和阴影：

1. 聚光灯：

a) 程序中设置了两个点光源，其中一个是聚光灯。

b) 聚光灯对其他表面的影响是通过其顶点来实现的。

c) 程序中将地板切割成了很多小块来实现较为柔和的光照。



程序效果图

2. 通过对光源和篮球位置的计算，使篮球在地板上产生阴影，有拉伸和渐变色，如图：



3. 阴影的大小和深度与篮球高度有关，篮球越接近地面，阴影越小越深。



球离地较远时的阴影

H. 观众席和观众:

1. 网上的确有现成的 **Staples Center** 的 3ds 模型，不过都是要收费的，而且贼贵贼贵的，所以我只能自己制作了简陋的观众席，只制作了低层的座椅和走道。
2. 椅子的颜色是参照了真实的 **Staples Center** 的座椅颜色的，篮板后最低的一批座椅为灰色，其他的座椅为蓝色。
3. 观众使用的是 **NBA Live2007** 里面的素材，一共弄了 4 种观众的图片，程序里面放了 4 个观众，是程序一开始独立随机选取的图片，要是一开始没能看全 4 张的话，请多试几次--

