

动态轮胎代码解读

User:

使用通俗易懂的语言解读这段动态轮胎代码：

```
//Animate the Wheels

const animWheel = new BABYLON.Animation("wheelAnimation", "rotation.y", 30,
BABYLON.Animation.ANIMATIONTYPE_FLOAT, BABYLON.Animation.ANIMATIONLOOPMODE_CYCLE);
const wheelKeys = [];
//At the animation key 0, the value of rotation.y is 0

wheelKeys.push({
    frame: 0,
    value: 0
});

//At the animation key 30, (after 1 sec since animation fps = 30) the value
// of rotation.y is 2PI for a complete rotation
wheelKeys.push({
    frame: 30,
    value: 2 * Math.PI
});

//set the keys
animWheel.setKeys(wheelKeys);
//Link this animation to a wheel
wheelRB.animations = [];
wheelRB.animations.push(animWheel);
scene.beginAnimation(wheelRB, 0, 30, true);
```

Yuanbao:

下面是用通俗易懂的语言解读这段动态轮胎代码：




核心作用

这段代码让一个叫 `wheelRB` 的轮胎模型在 1 秒内匀速旋转 360 度（一整圈），并持续循环转动。

通俗版分步解析




1. 创建动画规则

```
const animWheel = new BABYLON.Animation(...);
```

-  告诉程序：“我们要给轮胎创建动画，动画名字叫 `wheelAnimation`”
-  修改轮胎的**旋转角度**(`rotation.y` 表示绕垂直轴旋转)
-  每秒播放**30帧**（流畅动画的标准帧率）
-  动画类型是**数字变化**（因为旋转角度是数字）
-  播放模式是**无限循环**（轮胎会一直转下去）



2. 定义关键动作

```
wheelKeys.push({ frame: 0, value: 0 });  
wheelKeys.push({ frame: 30, value: 2 * Math.PI });
```

-  **第0秒**（帧0）：轮胎初始角度 = **0度**（未转动）
-  **第1秒**（帧30）：轮胎角度 = **360度**（用弧度制 `2π` 表示）
-  **关键点**：30帧 ÷ 30帧/秒 = **1秒完成一圈**




3. 绑定动画规则

```
animWheel.setKeys(wheelKeys);  
wheelRB.animations = [animWheel];
```

-  把时间轴规则(`wheelKeys`)注入动画
-  把动画绑定到叫 `wheelRB` 的轮胎对象上

4. 启动动画

```
scene.beginAnimation(wheelRB, 0, 30, true);
```

-  命令场景播放轮胎动画
-  从**第0帧**播放到**第30帧**（对应0秒到1秒）
-  最后一个参数 `true` 表示：播完**立即重播**（永远循环）

直白效果描述

 当程序运行时，轮胎会以**每1秒转1圈**的速度匀速旋转，就像汽车在平稳行驶时车轮转动的效果，并且会一直循环下去。

关键数值关系

通过线性插值，程序会自动计算中间帧的旋转角度（例如第15帧时恰好转半圈180°）

这段代码是游戏/3D应用中实现运动效果的典型方式，通过定义起始/结束状态，由系统自动生成中间过程。