食谱

此页面列出了一些"食谱"，它们将通过几个步骤教您如何完成常见任务。

介绍

想要快速开始使用 TopDown 引擎吗？ 此页面包含您可以在许多基本情况下遵循的简单步骤，没有解释，只是快速步骤。不要犹豫，查看文档的其余部分以了解有关引擎如何工作的更多信息！请注意，其中大多数将提供 TDE / Unity 版本，这只是为了提供它们最初编写的上下文。但可以安全地假设它们应该适用于 Unity 和 TDE 的任何更高版本。

创建一个基本的 3D 敌人

- 全新安装 2019.3.3f1
- 导入自顶向下引擎 v1.10.1
- 打开 Loft3D 演示场景
- 创建一个空的游戏对象，将其设置为 -9.5,1.5,0
- 创建一个立方体，将其比例设置为 2,2,2，将其设置在空的下方
- 将空重命名为"测试"
- 在测试中，添加一个字符组合
- 按 Autobuild AI Character 3D 按钮
- 在字符检查器中，将 CharacterModel 设置为 Cube
- 按下播放键，拿起旁边的霰弹枪，尽情享受吧

设置武器模型

- 打开上校演示场景

设置武器模型：

- 在其 HandR 节点下，创建一个空对象，将其命名为 MyWeaponModel
- 添加一个 WeaponModel 组合，将它的 WeaponID 设置为 MyWeapon
- 在它下面，创建一个新的立方体，移除它的盒子碰撞器，将它的比例设置为 0.003,0.003,0.003
- 在 MyWeapon 的 WeaponModel 检查器上，将立方体拖入 TargetModel 插槽中
- 按播放，注意方块消失了，这是正常的，它的武器没有装备

设置武器：

- 复制 Loft3D 的 AssaultRifle 预制件，将其命名为 MyWeapon
- 在其 ProjectileWeapon 的 ID 部分，将 WeaponName 设置为 MyWeapon
- 打开那个新的预制件，删除它的 ModelContainer 节点和它下面的所有东西，也删除反馈
- 清除其 ProjectileWeapon 的动画师和动画参数名称
- 在其顶层删除动画师
- 保存预制件，并在上校的 CharacterHandleWeapon 上，将其指定为它的 InitialWeapon
- 按播放键，你现在有一个不错的射击方块

创建一个基本的 2D 可玩角色

- 全新安装 2019.3.3f1
- 导入 TDE v1.10.1
- 打开 Minimal2D 演示场景
- 创建一个新的空游戏对象，将其放置在 40,-20,0 处，将其命名为 Test
- 创建一个新的精灵对象，将其精灵设置为 Adventurer_0，将图层排序为 Characters，将其嵌套在 Test 下，在 0,0,0
- 在测试中，添加一个字符组合
- 按 AutoBuild Player Character 2D
- 将 NewSprite 拖入 CharacterModel 插槽
- 在 CharacterOrientation2D 下，将面向模式设置为 MovementDirection，选中 Model Should Flip
- 打开考拉预制件，复制脚步节点
- 粘贴到你的场景中
- 将它嵌套在 Test 下，在 0,0,0
- 按播放

创建一个基本的 3D 可玩角色

- 全新安装 2019.3.3f1
- 导入 TDE v1.10.1
- 打开 MinimalScene3D 演示场景
- 创建一个新的空游戏对象，将其放置在 0,0,0 处，将其命名为 Test
- 创建一个新的立方体，将其拖到 Test 下，将其放置在 0,0,0 处，将其命名为 Model，移除其 BoxCollider
- 在测试中，添加一个字符组合
- 按 AutoBuild Player Character 3D
- 在 Character 组件上，将模型拖入 CharacterModel 插槽
- 在 CharacterOrientation3D 下，将 Rotation Mode 设置为 MovementDirection，将模型拖入 Movement Rotating Model 插槽
- 在您的项目面板中拖动该测试对象以使其成为预制件，将其从场景中移除
- 选择 LevelManager，将新的预制件拖到 PlayerPrefabs 的第一个数组条目中
- 按播放，享受你的新朋友

创建 IK 动力武器

- 在 Unity 2019.4.17f1 中创建一个新的空项目
- 导入自顶向下引擎 v1.10.1
- 打开 Loft3D 演示场景
- 在该场景中创建一个新的空游戏对象，将其命名为 CubeGun
- 创建一个新的立方体，将其嵌套在 CubeGun 节点下，重置其变换
- 将立方体的比例设置为 0.2,0.5,1.5
- 在 CubeGun 上，添加一个 ProjectileWeapon 组件
- 展开它的 Projectiles 部分，将 spawn offset 设置为 0,0,1
- 在 CubeGun 上，添加一个 MMSimpleObjectPooler，将 AssaultRifleBullet 预制件从 Project 视图拖到其 GameObjectToPool 插槽
- 在 CubeGun 上，添加一个 WeaponAim3D 组件，将其 AimControl 设置为 SecondaryThenPrimaryMovement，并将其武器旋转速度设置为 0
- 将 CubeGun 拖到您的项目面板中，使其成为预制件。
- 选择 Suspenders 预制件，并在其 CharacterHandleWeapon 组件中，将 CubeGun 拖动到它的 InitialWeapon 插槽中
- 按播放
- 您现在可以四处移动并使用正确定位的立方体武器进行射击

从那里，如果您想设置 IK，您需要执行以下操作：

- 确保在你的角色上正确设置了 IK（参见我之前链接的文档），在已经是这种情况的悬挂器上
- 打开 CubeGun 预制件
- 创建一个新的空游戏对象，将其嵌套在 CubeGun 下，将其命名为 Left，将其定位在 -0.15,0,-0.1
- 创建一个新的空游戏对象，将其嵌套在 CubeGun 下，将其命名为 Right，将其定位在 0.1,-0.1,-0.65
- 在 CubeGun 的 ProjectileWeapon 上，在 IK 部分下，将 Left 拖入 LeftHandHandle 插槽，将 Right 拖入 RightHandHandle 插槽
- 保存您的预制件，按播放
- 您角色的手现在已正确定位

创建一扇需要钥匙才能打开的门

在这个例子中，我们将创建一个"门"（技术上是一个消失的立方体），我们将它与一个需要钥匙才能打开的交互区域相关联。

- 在 Unity 2019.4.28f1 上全新安装 TDE v2.1
- 打开 Loft3D 演示场景

创建门

- 创建一个新的立方体，将其命名为"Door"，将其定位在 -15,0.5,0，将其比例设置为 1,5,1
- 创建一个新的立方体，将其命名为"DoorZone"，将其 BoxCollider 设置为触发，将其定位在 -14,0,0
- 向 DoorZone 对象添加一个 KeyOperatedZone，向 OnActivation 添加一个事件，将 Door 拖到其插槽中，然后选择 GameObject > SetActive（这里我们只是让门消失，但如果你有一扇带有实际 Open() 方法的门，那也是你绑定它的地方）
- 在其检查器的底部，将 RequiresKey 设置为 true，然后在 KeyID 字段中输入"MyKey"

创建密钥

- 在您的项目中，创建一个文件夹，将其命名为 Resources
- 在其中，右键单击，创建 > 更多山脉 > 自顶向下引擎 > 库存引擎密钥，将其命名为 MyKey
- 将其 ItemID 设置为 MyKey，将描述设置为"a key"，将图标设置为 Koala_Items_MetalKey，将 TargetInventoryName 设置为 SuitMainInventory
- 在 Loft3D 场景中，创建一个新球体，将其命名为"MyKeyPicker"，将其碰撞器设置为 IsTrigger，将其定位在 -9,1,0
- 向其添加一个 ItemPicker 组件，将我们创建的 MyKey 资产拖到其 Item 插槽中，添加一个 PickableItem，选中 DisableObjectWhenDepleted

测试

- 按播放键，不要抓住球体/键，然后走在我们"门"（我们的 DoorZone 对象）旁边的小立方体的顶部，按空格键，没有任何反应，因为我们没有所需的键
- 向右移动，走在球体/钥匙的顶部抓住它，按 I 打开你的库存，注意它现在在那里
- 走在"门"旁边的小立方体上，按空格键，注意门按预期消失了

让新的输入系统工作

- 全新安装最新的 Unity 稳定版 (2019.4.17)
- 导入 TDE v1.10.1
- 通过包管理器导入 InputSystem 1.0.1
- 打开 MinimalScene3D_InputSystem_Multiplayer 演示场景
- Unity 通常会丢失一些对新输入系统资产的引用，因此您可能需要重新绑定它们。这就像在任何项目中一样，在 TopDownEngineInputActions 中确保键盘和游戏手柄方案具有正确的绑定（键盘应该是键盘和鼠标，游戏手柄应该是游戏手柄）。
- 在 InputSystemManager_Player1（和 2）上，如果引用丢失，则绑定事件
- 按播放

用单独的输入设置两个字符

- 2019.4.19f1新建项目
- 导入 TDE v1.10.1
- 打开 MinimalScene3D 演示场景
- 选择 LevelManager，将其 PlayerPrefabs 数组大小设置为 2
- 选择 Suspenders 预制件，将其 PlayerID 设置为 Player2，将其拖动到 LevelManager 插槽 2（元素 1）中
- 选择 UICamera，复制 InputManager comp，将其粘贴为新的
- 在新粘贴的 InputManager 中，将 PlayerID 更改为 Player2
- 按播放，您现在可以使用 wasd 控制最小字符，带箭头的吊带

用弹药制造武器

- 在 Unity 2019.4.19f1 中创建一个新项目
- 导入 TDE v1.10.1
- 打开 KoalaDungeon 演示场景
- 创建一个新的空对象，将其命名为 TestWeapon
- 添加一个 ProjectileWeapon comp，在 Magazine 下，检查 MagazineBased，将 MagazineSize 设置为 10，在 Settings 下检查 InitializeOnStart
- 给它添加一个 MMSimpleObjectPooler 组合，在它的 GameObjectToPool 插槽中拖动一个 KoalaRifleBullet（对于这些步骤，我没有创建新弹药，它将使用步枪弹药）
- 添加一个 WeaponAmmo comp，将 AmmoID 设置为 KoalaRifleAmmo，AmmoInventoryName 设置为 KoalaMainInventory，Max Ammo 设置为 100，并确保选中 ShouldLoadOnStart

- 将 TestWeapon 对象拖到您的项目中以使用它制作预制件，然后将其从场景中移除
- 在资源文件夹中，右键单击 > MoreMountains > TopDownEngine > InventoryWeapon 以创建武器资产
- 将其命名为 TestWeapon，将其 ItemID 和 ItemName 设置为 TestWeapon，将其 TargetInventoryName 设置为 KoalaMainInventory，选中其 Equippable 复选框，将其 TargetEquipmentInventoryName 设置为 KoalaWeaponInventory，在其 EquippableWeapon 插槽中，拖动 TestWeapon 预制件，将其 ItemClass 设置为 Weapon
- 在场景中，选择 KoalaRiflePicker，并在其 ItemPicker comp 中，拖动 TestWeapon 资产
- 按播放键，将步枪和弹药选择器移到上方以抓住它们
- 按T，这将装备武器（它看起来并不多，我们没有设置它的视觉效果或任何东西），按E射击，做10次清空剪辑，按R重新加载，再做一次, 武器清空并按预期重新装满

创建预填充广告资源

- 在 Unity 2019.4.19f1 中创建一个新项目
- 导入 TDE v1.10.1
- 打开吊带预制件
- 在其 CharacterInventory 中，将 AutoPick 项目设置为 3：手枪弹药 10、突击步枪 1、突击步枪弹药 10
- 在 AutoEquipWeaponOnStart 中，拖动手枪
- 打开 Loft3D 演示场景，按播放，你现在手持手枪生成，按 T，你现在有一把突击步枪，再次按 T，现在它是一把手枪，重复你想要的次数

设置可损坏和不可损坏的射弹

- 在Unity 2019.4.17f1中新建一个项目
- 导入自顶向下引擎 v1.10.1
- 打开 MinimalScene3D 演示场景
- 选择 MinimalCharacter 预制件，在其 CharacterHandleWeapon 组件中，拖动 AssaultRifle 预制件
- 在场景中，选择 Border 节点（在 Level/Models/ 下），将其 y 值设置为 0.5
- 选择 AssaultRifleBullet 预制件，打开它
- 在其中你会发现两个现成的反馈，在 HitDamageable 一个中，添加一个 DebugLog 反馈，将其调试消息设置为"hit damageable"

- 在 HitNonDamageable 中，添加调试日志反馈，将其调试消息设置为"hit non damageable"
- 保存预制件，回到现场
- 创建一个新的立方体，将其位置设置为 3.5,1,-8，为其添加一个 Health 组件，将其图层更改为 Enemies
- 按播放键，向立方体射击，然后向墙壁射击，您应该会在控制台中看到类似的内容

设置武器模型启用程序

- 在全新安装的 Unity 2019.4.19f1 中，导入 TDE v1.10.1
- 打开考拉预制件，创建一个空子，命名为TestWeaponModel，添加一个 SpriteRenderer，将精灵设置为HornyBlue_0，排序图层角色，在图层1中排序，将其移动到0,2,0
- 添加一个动画师，将控制器设置为 HornsBlueAnimator
- 添加一个 MMAnimatorMirror 组件（它将把动画参数的状态从动画师 A 复制到动画师 B）。将其 SourceAnimator 设置为 KoalaModel 节点的动画制作器，并将 TargetAnimator 设置为其正上方的动画制作器组件
- 禁用 TestWeaponModel 节点
- 在 Koala 节点上，添加一个 WeaponModelEnabler，将 Bindings 的大小设置为 1，将 TestWeaponModel 拖入插槽中， WeaponAnimationID 为 10（任意唯一 ID）
- 选择 KoalaRifleAmmoBased 预制件，展开其 AnimationParametersName 面板，将 WeaponAnimationID 设置为 10
- 在同一个面板中，将 StartAnimationParameter 设置为 SwordAttack（触发对 Horns 动画师的攻击的参数）
- 打开 KoalaModel 的 KoalaAnimatorController 动画控制器，给它添加一个 SwordAttack bool 参数
- 打开KoalaDungeon演示场景，按下播放键，在出生点上方拿起突击步枪，出现武器模型（蓝色家伙）
- 射击，蓝色的家伙会攻击

这只是其中一种方法，你也可以有一个动画师（在这种情况下是考拉），你可以在其中为武器/蓝色家伙制作动画，这些动画总是会播放，但只有在武器出现时才可见模型启动器正在激活模型。

设置接触伤害

- 在 Unity 2019.4.22f1 中创建一个新的空项目
- 导入 TDE v1.10.1
- 打开 MinimalScene3D
- 创建一个新的立方体，将其放置在 -5,0-,5
- 设置它的盒子碰撞器来触发
- 添加一个 DamageOnTouch 组件，将它的 TargetLayerMask 设置为 Player，将它的 DamageCausedKnockbackForce 设置为 20,20,0
- 按下播放键，走向立方体，与它碰撞时会被击退

创建一个简单的、类似炮塔的旋转 AI

- 在Unity 2019.4.22f1中新建一个项目
- 导入 TDE v1.10.1（在自述文件中检查版本）
- 打开 MinimalScene3D
- 创建一个新的空游戏对象，将其命名为 Test
- 在它下面，创建一个新的立方体，命名为 Model
- 在 Test 节点上，添加一个 Character 组件，按下其 Autobuild AI Character 3D 按钮，将 Model 节点拖到 Character 的 CharacterModel 插槽中
- 在 CharacterOrientation3D 上取消选中 ShouldRotateToFaceWeaponDirection，选中 ForcedRotation，将模型绑定到 MovementRotatingModel 插槽中，将 RotationMode 设置为 MovementDirection，
- 在 Test 节点上，添加 AIBrain、AIActionDoNothing、AIDecisionDetectTargetRadius3D 和 AIActionRotateTowardsTarget3D 组件
- 在 AIDecisionDetectTargetRadius3D comp 上，将 TargetLayerMask 设置为 Player
- 在 AIActionRotateTowardsTarget3D 上，检查 LockRotationX
- 在 AIBrain 上，创建一个新状态，将其命名为 Wait，将 AIActionDoNothing 设置为其动作，将 AIDecisionDetectTargetRadius3D 作为其转换决策，TrueState > Rotate
- 创建第二个状态，将其命名为 Rotate，将其 Action 设置为 AIActionRotateTowardsTarget3D
- 按播放，享受你的小 AI！

创建一个简单的 AI 将跟随并朝向玩家旋转

- 在Unity 2019.4.28f1中新建一个项目
- 导入 TDE v2.1
- 打开 MinimalScene3D
- 创建一个空的游戏对象，将其命名为"MyChar"，将其定位在 0,0,0
- 创建一个立方体，将其放置在 0,0,0 处，将其嵌套在 MyChar 下，将其命名为 MyModel
- 将 Character comp 添加到 MyChar，按 AutoBuild AI Character 3D
- 在 CharacterOrientation3D 中，将 MyModel 拖动到 MovementRotatingModel 插槽中，将旋转模式设置为 MovementDirection
- 添加一个 AI 大脑，AIDecisionDetectTargetRadius3D、AIActionMoveTowardsTarget3D、AIActionRotateTowardsTarget3D
- 在大脑中，添加一个新状态，将其命名为 Detect，为其添加一个过渡，将决策设置为 DetectTargetRadius，将 TrueState 设置为 Move
- 添加一个新状态，将其命名为 Move，no Transition，将其 Actions 设置为 Move 和 Rotate to target
- 在检测目标半径中将 TargetLayerMask 设置为 Player
- 按下播放键，四处走动，AI跟随并朝目标旋转

设置由视觉驱动的 AI 的 2D 锥体

- 在 Unity 2019.4.25f1 中全新安装 TDE v1.10.1
- 打开考拉地牢
- 选择左边第一个忍者，右击，完全解压
- 将其重命名为 NinjaWithCone（或其他），并制作它的预制件
- 打开考拉预制件，选择它的 ConeOfVision 节点，复制它
- 打开 NinjaWithCone 预制件，将 ConeOfVision 粘贴到其中，将其与 NinjaModel 或 Feedbacks 放在同一级别（在顶层之下），将其放置在 0,0,0
- 在该预制件的顶层，删除 AIDecisionTargetRadius2D
- 添加MMConeOfVision2D，设置ObstacleMask为Obstacles，TargetMask为Player，将ConeOfVision拖入底部的VisionMeshFilter槽中
- 添加一个 AIDecisionDetectTargetConeOfVision2D，并在 AIBrain 中，将第一个 State 的转换决策更改为 AIDecisionDetectTargetConeOfVision2D
- 按播放键，走进忍者的锥体，注意到它以玩家为目标并开始攻击它

如果您希望锥体也移动，则奖励步骤：

- 添加一个 AIActionRotateConeOfVision2D
- 添加一个 AIActionMoveRandomly2D
- 在AIBrain的第一个状态下，将动作（AIActionDoNothing）替换为AIActionMoveRandomly2D，并添加第二个动作，并使其成为AIActionRotateConeOfVision2D

设置命中扫描武器

- 在 Unity 2019.4.29f1 上全新安装 TDE v2.2
- 打开考拉地牢
- 创建一个空的游戏对象，将其命名为 TestHitscan
- 为其创建一个空子项，将其命名为 Model，为其添加一个 SpriteRenderer，将其 sprite 设置为 Koala_Weapons_Rifle，SortingLayer:Above
- 在顶层对象（TestHitscan）上，添加一个 HitscanWeapon 组件，在 HitscanSpawn 下，将弹丸生成偏移设置为 0.8, .1,0
- 在 Hitscan 部分下，将 Mode 设置为 TwoD，将 HitscanTargetLayers 设置为 Enemies、Obstacles
- 添加一个 WeaponAim2D 组件，将其 AimControl 设置为 Mouse, WeaponRotationSpeed:0
- 将 TestHitscan 对象拖到您的项目中以使其成为预制件，然后将其从场景中移除
- 选择 Koala 预制件，将 TestHitscan 预制件拖到其 CharacterHandleWeapon 的 InitialWeapon 插槽中
- 按播放键，您现在使用的武器会在瞄准敌人/陶罐时沿直线伤害/摧毁它们（现在您可以开始处理它的反馈以使其感觉更好！）

设置角色模型切换能力

- 在 Unity 2019.4.29f1 上全新安装 TDE v2.1
- 打开 MinimalScene2D 演示场景
- 打开 MinimalCharacter2D 预制件
- 创建一个新的空游戏对象，命名为AltModel，给它添加一个SpriteRenderer，并将其Sprite设置为Adventurer_0，然后将其位置设置为0,0.4,0，并禁用该对象
- 在顶层 (MinimalCharacter2D) 添加 CharacterSwitchModel 能力
- 向其 CharacterModels 数组添加两个条目，将 MinimalCharacterModel 拖到其第一个槽，将 AltModel 节点拖到其第二个槽
- 保存预制件，在 MinimalScene2D 上按 play，您现在可以四处移动，按 P 切换模型

创建一个新的加载屏幕

- 在 Unity 2019.4.28f1 上全新安装 TopDown Engine v2.1
- 复制 LoadingScreen 演示场景，将其命名为 NewLoadingScreen
- 将它添加到您的构建设置（文件，构建设置，将您的 NewLoadingScreen 拖到"构建中的场景"面板中）
- 打开它，选择 Main Camera，在它的 Camera 组件上，让我们将背景颜色更改为某种红色（如果您愿意，您可以在场景中进行其他更改，但红色是我们将确保我们使用的方式正确的）
- 保存场景
- 打开 Minimal2DDoors1，在它的 LevelManager 中，将 LoadingSceneName 设置为 NewLoadingScreen
- 按播放键，向上/向右移动穿过绿色门，然后向右移动直到看到蓝色门，进入（按空格键），注意我们正在使用新创建的加载屏幕加载新场景