创建自己的游戏

本页介绍了如何在 TopDown 引擎的基础上创建自己的游戏，以及如何安全地更新。

介绍

自上而下引擎的构建从一开始就考虑到了**可扩展性**。当然，您可以直接使用现有的类，但您也可以创建新类或在现有类的基础上进行构建。本页介绍了一些您可以添加内容和功能以创建**您自己的游戏的方法**。

视觉和声音

一个非常简单的方法是用您自己的**视觉资产替换视觉资产**。这样做非常简单，因为您所要做的就是找到文件并将其替换为您的作品。对此没有什么可说的。Unity 良好实践适用（当然，您可以从该实践中转移，但如果您需要一些参考，您可以使用这些）。您应该查看这些指南，以确保您的资产在所有平台上按预期工作，并且您的项目保持可维护性。

如果您无法确定**特定精灵**在演示文件中的位置，您只需在场景视图中选择预制件，然后单击其精灵渲染器的 Sprite 字段即可。这将在层次视图中突出显示精灵。然后，您只需在层次视图中右键单击精灵，然后单击"在 Finder 中显示"（或在 Windows 上的资源管理器中显示）。你现在要做的就是用你自己的文件替换这个文件，然后再次按下播放键。

调整现有组件

资产中的所有组件都在构建时考虑了许多用例。从同一个组件中，您通常可以通过他们的检查器调整值并选择不同的行为，最终将使您的游戏真正脱颖而出。在开始编写新的代码之前，如果有的话，不妨看看现有的类似类，查看他们的 API 文档，并确保您不能通过一些设置更改来获得您想要的东西。

添加新类

也许仅仅添加视觉效果不足以让您创建您想要的游戏。有时你会发现你需要新的代码来为你的想法赋予生命。

一种方法是简单地创建新类。也许你想要一个片尾字幕场景，展示参与过你那个项目的每个人？现在引擎中没有这样的组件，引擎中也没有任何靠近它的东西。这通常是您想要创建自己的类的那种情况。它与任何事情都无关，你可以自由地做任何你想做的事。从您自己的类中，您仍然可以访问引擎的其余部分，例如 LevelManager 或 GameManager，获取对当前角色的引用等。

装配定义

作为一种良好的做法，引擎使用程序集定义。这很好地分离了代码，并缩短了编译时间。但这也意味着引擎脚本不会"看到"其程序集定义之外的任何脚本。这就是为什么鼓励扩展脚本而不是直接修改它们的原因。如果您真的想修改核心类而不是扩展它们，并且需要访问外部 API，您可以编辑程序集定义以添加对它们的访问（脏），或删除所有程序集定义（非常脏）。

文件夹结构

在您的项目中，您可以将 TopDown Engine 文件夹放在您想要的任何位置，它不会放在特定的位置。也就是说，最好不要将您自己的内容放在 TopDown Engine 文件夹中，因为例如，如果您更新它，您可能必须将其完全删除。相反，最好有这样的文件夹结构：

资产/

- *游戏/您将在其中放置所有资产、脚本等。*
- ThirdParty/*一个文件夹，您将在其中放置您计划使用的所有第三方插件/框架/资产*

- 自顶向下引擎/*整个 TDE 文件夹*
- 其他插件/

如果您决定扩展引擎的类，它们将进入您的 Game 文件夹（或可能是 Game/Scripts/TDE/ 子文件夹）。有了这样的结构，更新引擎变得容易，您可以以任何方式组织您的游戏文件夹和子文件夹，而不会破坏自上而下引擎或其他插件中的任何内容。

遗产

有时您会发现您希望某些现有类的行为有所不同。也许这walljump能力不工作*正是*像你想它。尽管我非常希望引擎能够匹配所有可能的游戏风格，但这是不可能的，有时必须做出选择。但好消息是，这也被考虑在内了！

引擎中的大多数（如果不是全部）方法都标记为虚拟，这意味着您可以轻松覆盖它们以满足您自己的需要。我强烈建议使用继承和 多态来重用和扩展尽可能多的代码。

无论如何，我强烈建议不要修改任何这些，而是**将引擎的脚本扩展到您自己的类中**。这样您在更新到下一个版本时就不会删除您的工作。