

1. 可见性

在 Bevy 中，Visibility 用于控制某物是否被渲染。如果你希望一个实体存在于世界中，但不显示出来，你可以将其隐藏。

```
commands
    .spawn(
        MaterialMesh2dBundle {
            mesh: meshes.add(Rectangle::new(200.,
200.)).into(),
            material:
materials.add(Color::Srgba(Srgba::RED)),
            transform: Transform::from_xyz(-400., 0., 0.),
            visibility: Visibility::Hidden,
            ..default()
        }
    );
```

2. 可见性组件

在 Bevy 中，可见性由多个组件表示：

- Visibility：用户控制的开关（在这里设置你想要的）
- InheritedVisibility：用于 Bevy 跟踪来自任何父实体的状态
- ViewVisibility：用于 Bevy 跟踪实体是否应该实际显示

任何表示游戏世界中可渲染对象的实体都需要包含这些组件。
Bevy 内置的大多数 Bundle 类型都包括它们。

如果你在创建自定义实体时不使用这些 Bundle，可以使用以下之一来确保不会遗漏：

- SpatialBundle：用于变换 + 可见性
- VisibilityBundle：仅用于可见性

```
commands.spawn((
    Mesh2dHandle(meshes.add(Circle::new(100.))),
    materials.add(Color::Srgba(Srgba::BLUE)),
    // SpatialBundle::default(),
    // TransformBundle::default(),
    Transform::default(),
    GlobalTransform::default(),
    // VisibilityBundle::default(),
    Visibility::default(),
    InheritedVisibility::default(),
    ViewVisibility::default(),
));
```

如果你没有正确添加这些组件（例如，手动添加了 Visibility 组件但忘记了其他组件，因为你没有使用 Bundle），你的实体将不会渲染！

3. Visibility

Visibility 是“用户控制的开关”。这是你为当前实体指定所需状态的地方：

- Inherited（默认）：根据父实体显示/隐藏
- Visible：始终显示实体，无论父实体如何
- Hidden：始终隐藏实体，无论父实体如何

如果当前实体有任何子实体且其可见性为 Inherited，当你将当前实体设置为 Visible 或 Hidden 时，它们的可见性将受到影响。

如果一个实体有父实体，但父实体缺少与可见性相关的组件，行为将如同没有父实体一样。

4. InheritedVisibility

InheritedVisibility 表示当前实体基于其父实体的可见性状态。

InheritedVisibility 的值应视为只读。它由 Bevy 内部管理，类似于变换传播。一个“可见性传播”系统在 PostUpdate 调度中运行。

如果你想读取当前帧的最新值，应将你的系统添加到 PostUpdate 调度中，并在 VisibilitySystems::VisibilityPropagate 之后排序。

```
fn print_triangle_iherited_visibility(triangle:
Query<&InheritedVisibility, With<MyTriangle>>) {
    let Ok(inherited_visibility) = triangle.get_single()
else {
    return;
};
    info!("triangle inherited visibility:
{inherited_visibility:?}");
}
```

```
print_triangle_iherited_visibility
    .after(VisibilitySystems::VisibilityPropagate)
```

5. ViewVisibility

ViewVisibility 表示 Bevy 关于是否需要渲染该实体的最终决定。

ViewVisibility 的值是只读的。它由 Bevy 内部管理。

它用于“剔除”：如果实体不在任何相机或光源的范围内，则不需要渲染，因此 Bevy 将其隐藏以提高性能。

每帧，在“可见性传播”之后，Bevy 将检查哪些实体可以被哪些视图（相机或光源）看到，并将结果存储在这些组件中。

如果你想读取当前帧的最新值，应将你的系统添加到 PostUpdate 调度中，并在 VisibilitySystems::CheckVisibility 之后排序。

```
fn print_triangle_view_visibility(triangle:
Query<&ViewVisibility, With<MyTriangle>>) {
    let Ok(view_visibility) = triangle.get_single() else {
    return;
};
    info!("triangle view visibility: {view_visibility:?}");
}

print_triangle_view_visibility
    .after(VisibilitySystems::CheckVisibility)
```