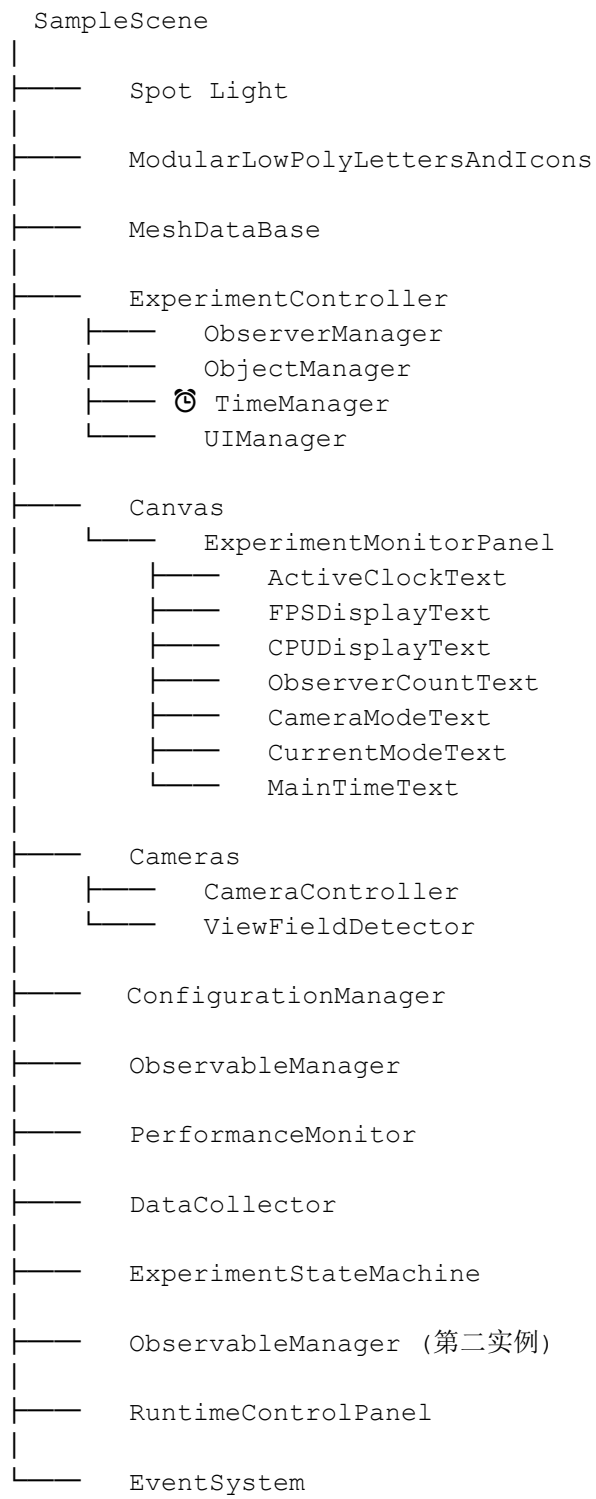
Unity 惰性更新实验架构搭建手册 - 完整版

场景层次结构 (Hierarchy)



核心组件详细配置

1. ExperimentController（实验总控制器）

Transform:

- Position: (0, 0, 0)
- Rotation: (0, 0, 0)
- Scale: (1, 1, 1)

ExperimentController (Script):

```
系统管理器引用:
├── Time Manager: TimeManager (Time Manager)
├── Object Manager: ObjectManager (Object Manager)
├── Observer Manager: ObserverManager (Observer Manager)
├── Observable Manager: ObservableManager (Observable Manager)
├── Configuration Manager: ConfigurationManager (Configuration
Manager)
├── State Machine: ExperimentStateMachine (Experiment State Machine)
├── UI Manager: UIManager (UI Manager)
├── Camera Controller: CameraController (Camera Controller)
├── Performance Monitor: PerformanceMonitor (Performance Monitor)
├── Runtime Control Panel: RuntimeControlPanel (Runtime Control
Panel)
└── Data Collector: DataCollector (Data Collector)

实验状态:
├── Current Mode: Traditional
└── Is Experiment Running: [根据运行状态]

调试模式设置:
├── Debug Mode Enabled: [可配置]
├── Debug Clock Count: 65536
├── Debug Add Clock Count: 500
└── Clock Spacing: 2.5
```

2. ObserverManager

ObserverManager (Script):

```
配置:
├── Observer Prefab: Observer
├── Experiment Controller: ExperimentController (Experiment
Controller)
├── Observer Spacing: 5
├── Observer Y Position: 5
├── Selected Color: [绿色]
└── Normal Color: [默认色]
```

3. ObjectManager

ObjectManager (Script):

基础配置：
├── Clock Prefab: Clock Variant
├── Clock Container: None (Transform)
└── Number Database: MeshDatabase (Clock Number Database)

生成配置：
├── Spacing: 4

累计信息（只读）：
├── Total Clock Count: 0
├── Active Clock Count: 0
└── Next Clock Start Time: 0

系统引用（可配）：
├── Time Manager: TimeManager (Time Manager)
├── Configuration Manager: ConfigurationManager (Configuration Manager)
├── UI Manager: UIManager (UI Manager)
├── Observer Manager: ObserverManager (Observer Manager)
└── View Field Detector: ViewFieldDetector (View Field Detector)

4. ⏰ TimeManager

TimeManager (Script):

主时间轴：
├── Is Experiment Running: [运行状态]
├── Is Paused: [暂停状态]
└── Main Time: 0

模式时间记录：
├── Traditional Mode Total Time: 0
├── Lazy Mode Total Time: 0
└── Current Mode Start Time: 0

模式切换历史：
└── Mode Switch History: [List is empty]

5. UIManager

UIManager (Script):

主要面板：
├── Configuration Panel: None (Game Object)
├── Ready Panel: None (Game Object)
├── Performance Panel: None (Game Object)
├── Comparison Panel: None (Game Object)
├── Warning Panel: None (Game Object)
├── Control Panel: None (Game Object)
└── Canvas: Canvas

运行时控制：
├── Runtime Control Panel: None (Runtime Control Panel)
├── Add Confirmation Dialog Panel: None (Game Object)
└── Confirmation Dialog Prefab: None (Game Object)

实验监视面板：
└── Experiment Monitor Panel: ExperimentMonitorPanel

核心监控显示：
├── Main Time Text: MainTimeText (Text Mesh Pro UGUI)
├── Active Clock Text: ActiveClockText (Text Mesh Pro UGUI)
├── Observer Count Text: ObserverCountText (Text Mesh Pro UGUI)
├── Current Mode Text: CurrentModeText (Text Mesh Pro UGUI)
├── Fps Display Text: FPSDisplayText (Text Mesh Pro UGUI)
├── Cpu Display Text: CPUDisplayText (Text Mesh Pro UGUI)
└── Camera Mode Text: CameraModeText (Text Mesh Pro UGUI)

模式切换控制：
├── Mode Switch Button: None (Button)
└── Pause Resume Button: None (Button)

对比面板组件：
├── Traditional FPS Text: None (Text Mesh Pro UGUI)
├── Traditional CPU Text: None (Text Mesh Pro UGUI)
├── Traditional Update Rate: None (Text Mesh Pro UGUI)
├── Lazy FPS Text: None (Text Mesh Pro UGUI)
├── Lazy CPU Text: None (Text Mesh Pro UGUI)
├── Lazy Update Rate Text: None (Text Mesh Pro UGUI)
├── Fps Gain Text: None (Text Mesh Pro UGUI)
├── Cpu Saved Text: None (Text Mesh Pro UGUI)
└── Efficiency Ratio Text: None (Text Mesh Pro UGUI)

其他 UI 组件：
├── Fps Bar: None (Image)
├── Cpu Bar: None (Image)
├── Pause Button: None (Button)
├── Switch Mode Button: None (Button)
├── Reset Button: None (Button)
├── Save Data Button: None (Button)
├── Pause Button Text: None (Text Mesh Pro UGUI)
├── Tooltip Text: None (Text Mesh Pro UGUI)
├── Confirmation Message: None (Text Mesh Pro UGUI)
├── Confirm Add Button: None (Button)
├── Cancel Add Button: None (Button)
├── Objects In View Text: None (Text Mesh Pro UGUI)
└── Pending Updates Text: None (Text Mesh Pro UGUI)

系统引用：
├── Time Manager: TimeManager (Time Manager)
├── Object Manager: ObjectManager (Object Manager)
├── Observer Manager: ObserverManager (Observer Manager)
├── Experiment Controller: ExperimentController (Experiment
Controller)

```
├── Performance Monitor: PerformanceMonitor (Performance Monitor)
└── Camera Controller: CameraController (Camera Controller)
```

6. Canvas

Canvas 组件:

- Render Mode: Screen Space - Overlay
- Pixel Perfect: ☐
- Sort Order: 0
- Target Display: Display 1
- Additional Shader Channels: Nothing

Canvas Scaler:

- UI Scale Mode: Constant Pixel Size
- Scale Factor: 1
- Reference Pixels Per Unit: 100

Graphic Raycaster:

- Ignore Reversed Graphics: ☑
- Blocking Objects: None
- Blocking Mask: Everything

7. ExperimentMonitorPanel

Rect Transform:

- Pos: (100, -100, 0)
- Size: (100, 100)
- Anchors: (0.5, 0.5)
- Pivot: (0.5, 0.5)
- Scale: (1, 1, 1)

Vertical Layout Group:

- Padding: 50
- Spacing: 50
- Child Alignment: Upper Left
- Control Child Size: Width ☑, Height ☑
- Use Child Scale: Width ☑, Height ☑
- Child Force Expand: Width ☑, Height ☑

8. UI Text 组件 (TextMeshPro)

所有 Text 组件的通用配置：

- Font Asset: LiberationSans SDF (TMP_Font Asset)
- Material Preset: LiberationSans SDF Material
- Font Style: B I U S ab AB SC

- Font Size: 36
- Auto Size: ☐
- Vertex Color: 白色
- Character Spacing: 0
- Word Spacing: 0

9. CameraController

Transform:

- Position: (0, 0, 10)
- Rotation: (0, -180, 0)
- Scale: (1, 1, 1)

Camera 组件:

- Clear Flags: Skybox
- Background: [天蓝色]
- Culling Mask: Everything
- Projection: Perspective
- FOV Axis: Vertical
- Field of View: 60
- Physical Camera: ☐
- Clipping Planes: Near 0.3, Far 10
- Viewport Rect: X 0, Y 0, W 1, H 1
- Depth: 0
- Rendering Path: Use Graphics Settings
- Target Texture: None (Render Texture)
- Occlusion Culling: ☑
- HDR: Use Graphics Settings
- MSAA: Use Graphics Settings
- Allow Dynamic Resolution: ☐
- Target Display: Display 1

CameraController (Script):

相机模式:
└── Current Mode: External Observer

移动控制:
├── Move Speed: 10
├── Speed Multiplier: 2
└── Move Smoothing: 0.1

缩放控制:
├── Zoom Speed: 0.1
├── Min Ortho Size: 5
└── Max Ortho Size: 50

视野检测设置:
├── Detection Interval: 0.1
└── Show Viewport Border: ☑

视觉反馈：
├── Viewport Border: None (Line Renderer)
├── Spectator Border Color: [绿色]
└── Observer Border Color: [绿色]

10. ViewFieldDetector

Transform:

- Position: (0, 0, 0)
- Rotation: (0, 0, 0)
- Scale: (1, 1, 1)

Box Collider:

- Edit Collider: [可编辑]
- Is Trigger: ☑
- Provides Contacts: ☐
- Material: None (Physic Material)
- Center: (0, 0, 10)
- Size: (0, 0, 0)
- Layer Overrides: [配置层级]

Rigidbody:

- Mass: 1
- Drag: 0
- Angular Drag: 0.05
- Automatic Center Of Mass: ☑
- Automatic Tensor: ☑
- Use Gravity: ☑
- Is Kinematic: ☐
- Interpolate: None
- Collision Detection: Discrete
- Constraints: [无约束]

View Field Detector (Script):

调试设置：
├── Show Debug Info: ☐
└── Detected Clock Color: [绿色]

11. ConfigurationManager

ConfigurationManager (Script):

当前配置状态：
└── Current State: [状态值]

预设配置：
└── Presets: 3

配置限制：
├── Max Clock Count: 10000
├── Max Observer Count: 10
└── Default Spacing: 2.5

系统引用：
├── Object Manager: ObjectManager (Object Manager)
├── Observer Manager: ObserverManager (Observer Manager)
├── UI Manager: UIManager (UI Manager)
└── State Machine: ExperimentStateMachine (Experiment State Machine)

12. PerformanceMonitor

PerformanceMonitor (Script):

监控设置：
├── Enable Monitoring: ☑
├── Update Interval: 0.5
├── Fps Sample Size: 60
└── Show Debug Info: ☐

性能阈值：
├── Low FPS Threshold: 30
├── High CPU Threshold: 80
└── High Memory Threshold: 1000

性能事件：
├── Enable Performance Events: ☑
├── Performance Events: 0
└── Max Event History: 100

CPU 监控设置：
├── Default Target FPS: 60
├── Base CPU Usage: 20
└── Cpu Smoothing Factor: 0.3

系统引用：
├── Data Collector: DataCollector (Data Collector)
└── UI Manager: UIManager (UI Manager)

13. DataCollector

DataCollector (Script):

数据收集配置：
├── Recording Duration: 10
└── Snapshot Interval: 0.1

当前状态：
├── Is Recording: ☐
├── Recording Progress: 0
├── Current Clock Count: 0

```
└─── Current Recording Mode: Traditional

系统引用:
├─── Experiment Controller: ExperimentController (Experiment
Controller)
├─── Performance Monitor: PerformanceMonitor (Performance Monitor)
├─── Object Manager: ObjectManager (Object Manager)
├─── UI Manager: None (UI Manager)
└─── Time Manager: TimeManager (Time Manager)
```

14. ExperimentStateMachine

ExperimentStateMachine (Script):

```
当前状态:
├─── Current State: Initialization
├─── Previous State: Initialization
└─── State Enter Time: 0

状态历史:
└─── State History: 0

状态事件:
├─── On State Changed (ExperimentState, ExperimentState)
│    └─── List is empty
│
├─── On Configuration Entered ()
│    └─── List is empty
│
├─── On Ready Entered ()
│    └─── List is empty
│
├─── On Running Entered ()
│    └─── List is empty
│
├─── On Paused Entered ()
│    └─── List is empty
│
└─── On Data Collection Entered ()
     └─── List is empty

系统引用:
├─── UI Manager: UIManager (UI Manager)
├─── Configuration Manager: ConfigurationManager (Configuration
Manager)
├─── Experiment Controller: ExperimentController (Experiment
Controller)
└─── Performance Monitor: PerformanceMonitor (Performance Monitor)
```

15. ObservableManager (第二实例)

ObservableManager (Script):

更新统计：
├─── Frame Update Count: 0
├─── Total Update Count: 0
├─── Last Update Time: 0
└─── Registered Object Count: 0

性能设置：
├─── Max Updates Per Frame: 100
└─── Enable Update Throttling: ☑

系统引用：
├─── Object Manager: ObjectManager (Object Manager)
├─── UI Manager: UIManager (UI Manager)
└─── Performance Monitor: PerformanceMonitor (Performance Monitor)

16. RuntimeControlPanel

RuntimeControlPanel (Script):

面板控制：
├─── Panel Root: None (Game Object)
└─── Overlay Background: None (Game Object)

时钟调整：
├─── Clock Adjust Input: None (TMP_Input Field)
├─── Add Clocks Button: None (Button)
├─── Remove Clocks Button: None (Button)
├─── Current Clock Count Text: None (Text Mesh Pro UGUI)
└─── Next Clock Time Text: None (Text Mesh Pro UGUI)

快速按钮：
├─── Add 100 Button: None (Button)
├─── Add 500 Button: None (Button)
├─── Add 1000 Button: None (Button)
├─── Remove 100 Button: None (Button)
└─── Remove 500 Button: None (Button)

观测者调整：
├─── Observer Adjust Input: None (TMP_Input Field)
├─── Add Observers Button: None (Button)
├─── Remove Observers Button: None (Button)
└─── Current Observer Count Text: None (Text Mesh Pro UGUI)

控制按钮：
├─── Continue Experiment Button: None (Button)
├─── Apply Changes Button: None (Button)
└─── Cancel Button: None (Button)

状态显示：
├─── Status Text: None (Text Mesh Pro UGUI)
└─── Pending Changes Text: None (Text Mesh Pro UGUI)

系统引用：

```
├─── Configuration Manager: ConfigurationManager (Configuration
Manager)
├─── Object Manager: ObjectManager (Object Manager)
├─── Observer Manager: ObserverManager (Observer Manager)
├─── Experiment Controller: ExperimentController (Experiment
Controller)
├─── State Machine: ExperimentStateMachine (Experiment State Machine)
└─── UI Manager: UIManager (UI Manager)
```

搭建步骤详解

Step 1: 基础场景配置

1. 创建新 Unity 场景 (Unity 2022.3.6f1)
2. 添加 Spot Light 光源
3. 设置场景背景色

Step 2: 核心控制器构建

1. 创建 ExperimentController 空对象
2. 添加 ExperimentController 脚本
3. 创建四个子对象并添加对应脚本：
    - ObserverManager → ObserverManager.cs
    - ObjectManager → ObjectManager.cs
    - TimeManager → TimeManager.cs
    - UIManager → UIManager.cs

Step 3: UI 系统搭建

1. 创建 Canvas (自动创建 EventSystem)
2. 设置 Canvas 为 Screen Space - Overlay
3. 创建 ExperimentMonitorPanel 子面板
4. 添加 Vertical Layout Group 组件
5. 创建七个 TextMeshPro 子对象

Step 4: 相机系统配置

1. 创建 Cameras 父对象
2. 添加 CameraController 和 Camera 组件
3. 创建 ViewFieldDetector 子对象
4. 添加 Box Collider (Is Trigger)和 Rigidbody
5. 配置 ViewFieldDetector 脚本

Step 5: 管理器组件设置

1. 创建各独立 Manager 对象
2. 添加对应脚本组件

3. 在 ExperimentController 中建立引用关系

Step 6: 引用关系配置

1. 将所有 Manager 拖拽到 ExperimentController 的对应字段
2. 将 UI Text 组件拖拽到 UIManager
3. 配置各 Manager 之间的交叉引用

关键配置要点

必要设置

1. **ViewFieldDetector 的 Collider 必须设为 Is Trigger**
2. **所有 Text 组件必须使用 TextMeshPro**
3. **CameraController 需要同时有 Camera 组件**
4. **ExperimentController 必须正确引用所有 Manager**

性能优化建议

- ObservableManager 的 Max Updates Per Frame 设为 100
- Enable Update Throttling 保持开启
- Detection Interval 设为 0.1 秒

调试技巧

- Debug Mode Enabled 开启可看到详细日志
- Show Debug Info 显示视野检测信息
- Show Viewport Border 可视化检测范围

Clock Variant 预制体详细配置

Clock Variant 预制体结构

Layer: TransparentFX
Prefab 来源: Clock Free Asset (Asset Store)
Asset 链接: https://assetstore.unity.com/packages/3d/props/interior/clock-free-44164

```
  Clock Variant
|
├────   Face (钟面)
|    ├────   HourBoard (时钟板)
|    |    └──── DigitalHour (数字小时显示)
|    |         ├──── HourLeft
|    |         └──── HourRight
|    |
|    └────   MinBoard (分钟板)
|         └──── DigitalMin (数字分钟显示)
|              ├──── MinLeft
```

```
│            └─── MinRight
│
├──    SecBoard (秒钟板)
│    └─── DigitalSec (数字秒显示)
│        ├─── SecLeft
│        └─── SecRight
│
└──    Arrow (指针组)
     ├─── HourHand (时针)
     ├─── MinuteHand (分针)
     ├─── Nail (中心钉)
     └─── SecondHand (秒针)
```

Clock Variant 根对象配置

Transform:

- Position: (0, 0, 0)
- Rotation: (0, 0, 0)
- Scale: (10, 10, 10)

Sphere Collider:

- Is Trigger: ☑
- Center: (0, 3.576279e-08, -0.000977242)
- Radius: 0.1785129
- Layer Overrides: [无]

Clock 脚本组件配置

```
时钟基础信息（共享）：
├─── Clock Id: -1
├─── Initial Time In Seconds: 0
└─── Current Mode: Traditional

传统模式数据集：
└─── Traditional State
     ├─── Hour: 0
     ├─── Minute: 0
     └─── Second: 0

惰性模式显示（基于主时间计算）：
├─── Hour: 0
├─── Minute: 0
└─── Second: 0

惰性模式状态（从 Observable Record State 派生）：
└─── Lazy State
     ├─── Hour: 0
     ├─── Minute: 0
     └─── Second: 0
```

显示时间：
```
├─── Hour: 0
├─── Minute: 0
└─── Second: 0
```

Observable Record State 缓存：
```
├─── Current State Second: 0
├─── Display Seconds: 0
├─── Last Observe Time: 0
└─── Time Elapsed: 0
```

函数状态：
```
├─── Has Evolution Function: ☐
└─── Has Apply Function: ☐
```

视觉组件：
```
├─── Pointer Seconds: SecondHand
├─── Pointer Minutes: MinuteHand
└─── Pointer Hours: HourHand
```

数字显示网格：
```
├─── Hourleft: HourLeft (Mesh Filter)
├─── Hourright: HourRight (Mesh Filter)
├─── Minuteleft: MinLeft (Mesh Filter)
├─── Minuteright: MinRight (Mesh Filter)
├─── Secondleft: SecLeft (Mesh Filter)
└─── Secondright: SecRight (Mesh Filter)
```

运行状态：
```
├─── Is Active: ☐
└─── Experiment Started: ☐
```

子对象详细配置

Face（钟面）

Transform: Position (0, 0, 0)
Mesh Filter: Clock2
Mesh Renderer: Material - Clock

HourBoard（时钟板）

Transform: Position (0.05, 0.03, 0)
Mesh Filter: Cube
Mesh Renderer: Material - Default-Material

HourHand（时针）

Transform: Position (5.627e-08, 7.848e-06, 0.1), Scale (1000, 1000, 1000)
Mesh Filter: HourHand
Mesh Renderer: Material - ArrowWhite

MinuteHand（分针）

Transform: Position (5.620e-08, 6.392e-05, 0.1), Scale (1000, 1000, 1000)
Mesh Filter: MinuteHand
Mesh Renderer: Material - ArrowWhite

SecondHand（秒针）

Transform: Position (4.307e-07, -1.716e-05, 0.1), Scale (1000, 1000, 1000)
Mesh Filter: SecondHand
Mesh Renderer: Material - ArrowWhite

Nail（中心钉）

Transform: Position (4.862e-07, 4.133e-07, 0)
Mesh Filter: Nail
Mesh Renderer: Material - Default-Material

关键技术要点

1. **Layer 设置**: 所有时钟对象都在 TransparentFX 层，用于特殊渲染处理
2. **Collider 配置**: 使用 Sphere Collider 作为触发器检测观测
3. **缩放因子**: 根对象缩放 10 倍，指针缩放 1000 倍确保正确显示
4. **材质系统**: 使用 Clock、ArrowWhite、Default-Material 等材质
5. **数字显示**: 通过替换 Mesh Filter 的 mesh 来显示不同数字

时钟更新机制

传统模式:

- 每帧直接更新所有时钟的显示
- 实时计算并旋转指针

惰性模式:

- 仅在被观测时更新
- 使用 Observable Record State 缓存状态
- 通过演化函数计算当前时间

Observer 预制体详细配置

- **用途**：作为实验中的内部观测者，模拟 NPC 或其他游戏内实体的“感知”行为，其射线检测是触发惰性更新的核心机制。
- **Layer**: Default

Observer 根对象配置

- **Transform**:

    - Position: (0.32, 4.2, 0)
    - Rotation: (0, 0, 180)
    - Scale: (5, 5, 5)

- **Icon.011 (Mesh Filter)**:
    - Mesh: Icon.011

- **Mesh Renderer**:
    - Materials: Element 0 - Default-Material

    - **Lighting**:
        - Cast Shadows: On
        - Receive Shadows: ☑
        - Contribute Global Illumination: ☐

    - **Probes**:
        - Light Probes: Blend Probes
        - Reflection Probes: Blend Probes

    - Anchor Override: None (Transform)

    - **Additional Settings**:
        - Motion Vectors: Per Object Motion

        - Dynamic Occlusion: ☑

- **Box Collider**:
    - Edit Collider: [可编辑]

    - Is Trigger: ☑

    - Provides Contacts: ☐

    - Material: None (Physic Material)

    - **Center: (0.003613681, 0.01899362, 0)**
    - **Size: (0.3182663, 0.4485726, 0)**
    - Layer Overrides: [无]

Observer Controller 脚本配置

- **移动设置**:
    - Move Speed: 5

- **旋转设置**:
    - Rotation Speed: 120

- **射线设置**:

    - Ray Distance: 20
      Ray Color: [绿色]
    - Ray Hit Color: [黄色]
    - Ray Width: 0.1
    - Last Raycast Time: 0

- **检测信息（只读）**:
    - Detected Clock Count: 0
    - Detected Clock Names: [List] 0
    - Detection Status: 未检测

运行时控制热键

```
Z - 生成时钟
M - 生成观测者
X - 启动实验
G - 开始数据采集
B - 切换更新模式（传统/惰性）
C - 添加更多时钟
Tab - 切换相机模式
P - 暂停/继续
R - 重置实验
```

本手册对应 Unity 版本: 2022.3.6f1 完整项目包含所有必要脚本文件 时钟资源来自 Unity Asset Store: Clock Free