



# Moon Deception - Unity Setup Guide

---

## Prerequisites

- **Unity 2022.3 LTS** or newer
  - **Universal Render Pipeline (URP)** — already configured
  - Basic Unity knowledge
- 



## Quick Start

### Option A: Automatic Setup

1. Open the project in Unity
2. Go to menu: **Moon Deception > Setup Phase 1**
3. Press **Play** to test

### Option B: Manual Setup

Follow the detailed steps below.

---



## Manual Setup Steps

### Step 1: Player Setup

#### 1. Create Player GameObject

- `GameObject > Create Empty` → name it `Player`
- **⚠️ IMPORTANT: Position:** `(0, 2, 0)` — Player must spawn ABOVE ground, not inside it!
- The CharacterController center is at `(0, 1, 0)`, so total player center will be at `Y=3`

#### 2. Add Visible Mesh to Player **⚠️ NEW**

- `GameObject > 3D Object > Capsule` as **child** of Player
- Name it `PlayerMesh`
- Local Position: `(0, 1, 0)` — aligned with CharacterController center
- Remove the Capsule's default `CapsuleCollider` component (Player uses CharacterController)
- This makes the player visible in the scene

#### 3. Add Components to Player

- `Add Component > Character Controller`
  - Height: `2`
  - Radius: `0.5`
  - Center: `(0, 1, 0)`
  - **Skin Width:** `0.08` **⚠️** Critical for ground collision!
  - **Step Offset:** `0.3`
  - Slope Limit: `45`
  - Min Move Distance: `0.001`

- Add Component > PlayerMovement
- Add Component > PlayerShooting
- Add Component > StressSystem
- Add Component > SimpleCrosshair (for visible crosshair)

#### 4. Setup Camera

- Drag Main Camera as child of Player
  - Camera Position: (0, 1.6, 0) (eye level)
  - Camera Rotation: (0, 0, 0)
  - In PlayerMovement, assign the camera to cameraTransform
- 

## Step 2: Layer Configuration

### 1. Create Layers (Edit > Project Settings > Tags and Layers)

- Layer 8: NPC
- Layer 9: Alien
- Layer 10: Environment

### 2. Configure PlayerShooting

- Set hitLayers to include: NPC, Alien, Environment
- 

## Step 3: Test Environment

### 1. Create Ground

- GameObject > 3D Object > Plane
- Scale: (10, 1, 10)
- Add a material for visibility
- Layer: Environment

### 2. Create Test NPC

- GameObject > 3D Object > Capsule → name it TestNPC
  - Add NPCBehavior script
  - Layer: NPC
  - Duplicate a few times and spread around
- 

## Step 4: GameManager Setup

### 1. Create GameManager

- GameObject > Create Empty → name it GameManager
  - Add GameManager script
  - Assign player reference
- 

## Step 5: UI Setup (Stress Bar)

### 1. Create Canvas

- GameObject > UI > Canvas
- Render Mode: Screen Space - Overlay

## 2. Create Stress Bar

- Under Canvas: UI > Slider → name it StressBar
- Anchor: Top-left
- Position: (120, -30, 0)
- Width: 200, Height: 20
- Uncheck Interactable

## 3. Style the Bar

- Background: Dark gray
- Fill: Red gradient (low=green, high=red)
- Delete Handle

## 4. Connect to StressSystem

- Select Player
  - In StressSystem, assign the Slider to stressSlider
- 

# Testing Checklist

## Movement Tests

- [ ] WASD moves player correctly
- [ ] Mouse look works (horizontal + vertical)
- [ ] Vertical look is clamped (-90° to 90°)
- [ ] Space bar makes player jump
- [ ] Gravity pulls player down
- [ ] Cursor is locked and hidden

## Shooting Tests

- [ ] Left-click fires raycast
- [ ] Debug rays visible in Scene view (yellow=miss, red=hit)
- [ ] Console shows hit messages with target name
- [ ] Hitting NPC triggers damage (if IDamageable)

## Stress System Tests

- [ ] Stress bar visible in UI
- [ ] AddStress(float) increases bar
- [ ] ReduceStress(float) decreases bar
- [ ] Stress clamped between 0-100
- [ ] Reaching 100 triggers OnStressMaxed event
- [ ] Passive recovery works when below threshold

## NPC Tests

- [ ] NPCs patrol between waypoints
- [ ] NPCs are on correct layer
- [ ] NPCs can receive damage

## GameManager Tests

- [ ] Game starts in Playing state

- [ ] Stress max triggers `Chaos` phase
  - [ ] Win/Lose conditions trigger correctly
- 

## Common Issues

### Player falls through floor / feels inside ground

- Ensure ground has a `Collider` component
- Check `CharacterController` height and center
- **⚠ Set Player Y position to 2 or higher** — The `CharacterController` center is offset by (0, 1, 0), so if Player is at Y=0, the feet will be at Y=0 (inside ground)
- **CRITICAL: Set Skin Width to 0.08** — Default 0.01 is too small and causes clipping!
- **Set Step Offset to 0.3** — Must be less than Height/2
- If using the scene from the repo, select Player in hierarchy and set Transform Position Y = 2

### Camera feels like TPS / orbits around player

- Camera must be a **child** of the Player GameObject
- Camera local position should be (0, 1.6, 0) — at eye level
- Camera local rotation must be (0, 0, 0) initially
- **Camera should NOT have any rotation scripts** — only `PlayerMovement` controls it
- In `PlayerMovement`, the camera only rotates on LOCAL X axis (pitch), never Y
- The player body rotates on Y axis (yaw) for horizontal mouse movement

### No crosshair visible

- Add the `SimpleCrosshair` script to any GameObject (e.g., Player or MainCamera)
- The script draws a crosshair using OnGUI, no Canvas needed

### Mouse look not working

- Verify `cameraTransform` is assigned in `PlayerMovement`
- Check if another script is controlling cursor

### Shooting doesn't hit anything

- Verify `hitLayers` includes target layers
- Ensure targets have `Collider` components
- Check raycast range in `PlayerShooting`

### Stress bar not updating

- Ensure `stressSlider` is assigned in `StressSystem`
- Check Slider min/max values (should be 0-1 for normalized)
- The updated `StressSystem.cs` auto-finds the slider by name "StressBar" if not assigned
- Check console for "[StressSystem] Initialized. Slider found: true/false"

### NPCs don't move/patrol

- NPCs now have **Auto Patrol** enabled by default (no waypoints needed)
- They patrol randomly within `patrolRadius` (default 5m) from start position
- Check console for "[NPC] name state: Idle -> Walking" logs
- Verify NPCs have `NPCBehavior` script attached

- If adding waypoints manually, ensure waypoint Transforms are assigned in inspector
- 

## Play Test Procedure

1. **Enter Play Mode** (Ctrl+P)
  2. **Test Movement:** Walk around, jump on objects
  3. **Test Shooting:** Aim at NPCs, check console for hits
  4. **Test Stress:** Call `player.GetComponent<StressSystem>().AddStress(20)` in console
  5. **Test Stress Max:** Add stress until 100, verify chaos phase triggers
- 

## Script Dependencies

```

GameManager
└ StressSystem (listens to OnStressMaxed)

PlayerMovement
└ CharacterController (required)
└ Camera (child transform)

PlayerShooting
└ Camera.main
└ IDamageable targets

StressSystem
└ UI Slider (optional)

NPCBehavior
└ NavMeshAgent (optional, for advanced AI)
└ IDamageable interface

```

## Phase 2: Alien Setup

### Alien Player Setup

1. **Create Alien GameObject**
  - `GameObject > 3D Object > Capsule` → name it `Alien`
  - Position: `(5, 2, 5)` — away from astronaut spawn
  - Tag: `Alien`
  - Layer: `Alien`
2. **Add Components to Alien**
  - Add Component > `AlienController` (TPS movement + camera)
  - Add Component > `HungerSystem` (hunger mechanics)
  - Add Component > `AlienEatSystem` (eating NPCs/aliens)
3. **Camera Setup for Alien**
  - Create a NEW camera for the Alien (or disable Main Camera when testing alien)

- Assign camera to `AlienController > cameraTransform`
- The TPS camera follows behind automatically — NO need to parent it

#### 4. Configure AlienEatSystem

- `Edible Tags : NPC , Alien`
- `Player Tag : Player` (NEVER edible)
- `Detect Range : 3`
- `Detect Radius : 1`

## Hunger Bar UI Setup

### 1. Create Hunger Slider

- Under Canvas: `UI > Slider` → name it `HungerBar`
- Anchor: Top-left
- Position: `(120, -60, 0)` — below stress bar
- Width: `200`, Height: `20`
- Uncheck `Interactable`

### 2. Style the Hunger Bar

- Background: Dark gray
- Fill: Green (full) → Red (empty) gradient
- Delete Handle

### 3. Auto-connection

- `HungerSystem` auto-finds `HungerBar` by name

## Eat Prompt UI

- `EatPromptUI` auto-creates the prompt panel if not found
- Shows “Press E to EAT” when looking at valid target

## Target Highlighting

- `TargetHighlight` auto-applies red emission glow to edible targets
- No manual setup needed — added dynamically

## Important Tags Setup

1. Go to `Edit > Project Settings > Tags and Layers`
2. Add Tag: `NPC` (for edible civilians)
3. Add Tag: `Alien` (for other alien players)
4. Add Tag: `Player` (astronaut — NEVER edible)

## GameController Setup (Input Conflict Fix)

**CRITICAL: To prevent both Player and Alien from moving together:**

### 1. Create GameController

- `GameObject > Create Empty` → name it `GameController`
- Add Component > `GameController`
- Assign `playerMovement` (drag Player GameObject)
- Assign `alienController` (drag Alien GameObject)

### 2. Switch Key: Press **TAB** to toggle between Player and Alien control

### 3. Alternative: Manual Disable

- When testing Alien: Disable `PlayerMovement` script on Player
- When testing Player: Disable `AlienController` script on Alien

## Testing Alien Mechanics

- [ ] TPS camera follows behind alien
  - [ ] WASD moves alien relative to camera direction
  - [ ] Mouse rotates camera smoothly
  - [ ] Hunger bar visible and decreasing over time
  - [ ] Looking at NPC shows red highlight (NOT self!)
  - [ ] “Press E to EAT” prompt appears
  - [ ] Pressing E destroys NPC and restores hunger
  - [ ] Blood decal spawns at eat location
  - [ ] CANNOT eat player (no highlight, no prompt)
  - [ ] CANNOT eat self (raycast ignores own collider)
  - [ ] TAB switches between Player and Alien control
  - [ ] Only ONE character moves at a time
- 

## Next Steps (Phase 3)

1. Multiplayer networking setup
  2. Voting system implementation
  3. Emergency meeting mechanics
  4. Map expansion and tasks
  5. Win/lose conditions refinement
- 

Happy developing! 