==

GRAVITY SHIFT - AUTOMATED PROJECT SETUP TOOL GUIDE

Project: Gravity Shift Battle Student: Xiangfeng Ding Course: CMP-6056B/CMP-7042B Game Development Institution: University of East Anglia (UEA) Unity Version: 2022.3.17f1 LTS

==

1. TOOL OVERVIEW
   ==

The Automated Project Setup Tool is a custom Unity Editor extension designed to streamline the development workflow and ensure consistency across the project.

PURPOSE:

- Automate repetitive manual setup tasks
- Eliminate human error in asset configuration
- Ensure correct component references and dependencies
- Maintain consistent project organization standards
- Accelerate development iteration cycles

BENEFITS:

1. Development Efficiency: Reduces setup time from hours to minutes
2. Quality Assurance: Eliminates manual configuration errors
3. Consistency: Ensures all assets follow the same standards
4. Reproducibility: Setup can be repeated reliably at any time
5. Documentation: Tool serves as living documentation of project structure

==

1. TOOL ARCHITECTURE
   ==

The tool consists of four main modules:

MODULE 1: ProjectSetupTool.cs (Main Controller)

- Provides user interface in Unity Editor
- Coordinates execution of all setup modules
- Logs progress and errors
- Accessible via: Tools > Gravity Shift > Complete Project Setup

MODULE 2: MaterialSetup.cs (Material Generator)

- Creates 10 materials with proper colors and properties
- Materials include: Player, Crystal, Ground, Wall, Checkpoint, Enemy, Barrier (transparent), Platform, Hazard, ExitPortal
- Configures transparency and rendering modes automatically

MODULE 3: PrefabSetup.cs (Prefab Generator)

- Creates 9 game prefabs with all required components
- Prefabs include: Player, Crystal, Checkpoint, Enemy, EnergyBarrier, MovingPlatform, PressurePlate, HazardZone, ExitPortal
- Automatically attaches scripts and configures components
- Applies materials to visual elements

MODULE 4: SceneSetup.cs (Scene Populator)

- Configures all 6 scenes (MainMenu + 5 levels)
- Places game objects in appropriate positions
- Creates level geometry (platforms, walls, hazards)
- Instantiates prefabs with correct placement

MODULE 5: UISetup.cs (UI Generator)

- Creates main menu interface
- Generates in-game HUD elements
- Builds pause menu system
- Implements multilingual language selector (EN/CN/JP/KR)

==

1. HOW TO USE THE TOOL
   ==

STEP 1: Open the Project in Unity

1. Launch Unity Hub
2. Click “Add” and select the GravityShift folder
3. Open the project with Unity 2022.3.17f1 LTS
4. Wait for Unity to import all assets

STEP 2: Run the Setup Tool

1. In Unity Editor menu bar, click: Tools > Gravity Shift > Complete Project Setup
2. A window titled “Project Setup Tool” will appear
3. Read the information about tool benefits
4. Click the “Run Complete Setup” button

STEP 3: Monitor Progress

The tool will execute four steps automatically:

- Step ¼: Creating materials… ✓
- Step 2⁄4: Creating prefabs… ✓
- Step ¾: Setting up scenes… ✓
- Step 4⁄4: Creating UI elements… ✓

Progress is displayed in:

- Setup Tool window (scroll view at bottom)
- Unity Console window (detailed logs)

STEP 4: Verify Completion

When setup completes, you will see:

- “Setup Complete!” dialog box

- “Project is ready to run. Press Play to test.” message
- All assets visible in Project window

STEP 5: Test the Game

1. Open any level scene (Assets/Scenes/Level1_Tutorial.unity)
2. Click the Play button in Unity Editor
3. Test player movement, gravity switching, and game mechanics

==

1. INDIVIDUAL MODULE EXECUTION
   ==

The tool also allows running individual modules separately:

OPTION 1: Create Materials Only

Click: “1. Create Materials” button Result: 10 materials created in Assets/Materials/

OPTION 2: Create Prefabs Only

Click: “2. Create Prefabs” button Result: 9 prefabs created in Assets/Prefabs/ Note: Requires materials to exist first

OPTION 3: Setup Scenes Only

Click: “3. Setup Scenes” button Result: All 6 scenes populated with game objects Note: Requires prefabs to exist first

OPTION 4: Create UI Only

Click: “4. Create UI Elements” button Result: UI elements added to all scenes Note: Requires scenes to be set up first

==

1. WHAT GETS CREATED
   ==

MATERIALS (Assets/Materials/)

1. PlayerMaterial.mat - Blue (0.2, 0.5, 1.0)
2. CrystalMaterial.mat - Cyan (0.0, 1.0, 1.0)
3. GroundMaterial.mat - Gray (0.5, 0.5, 0.5)
4. WallMaterial.mat - Dark Gray (0.3, 0.3, 0.3)
5. CheckpointMaterial.mat - Green (0.2, 1.0, 0.2)
6. EnemyMaterial.mat - Red (1.0, 0.2, 0.2)
7. BarrierMaterial.mat - Yellow Transparent (1.0, 1.0, 0.0, 0.5)
8. PlatformMaterial.mat - Brown (0.6, 0.4, 0.2)
9. HazardMaterial.mat - Dark Red (0.8, 0.1, 0.1)
10. ExitPortalMaterial.mat - Purple (0.8, 0.2, 1.0)

PREFABS (Assets/Prefabs/)

1. Player.prefab
    - Components: CharacterController, PlayerController, GravityController, PlayerEnergy
    - Child: PlayerBody (Capsule), PlayerCamera (Camera + AudioListener)
2. Crystal.prefab
    - Components: CrystalPickup
    - Visual: Rotating cube with trigger collider
3. Checkpoint.prefab
    - Components: Checkpoint
    - Visual: Green cylinder platform with trigger
4. Enemy.prefab
    - Components: EnemyAI
    - Visual: Red sphere with trigger collider
5. EnergyBarrier.prefab
    - Components: EnergyBarrier
    - Visual: Yellow transparent wall
6. MovingPlatform.prefab

    - Components: MovingPlatform
    - Visual: Brown platform

7. PressurePlate.prefab
    - Components: PressurePlate
    - Visual: Flat cylinder with trigger
8. HazardZone.prefab
    - Components: HazardZone
    - Visual: Dark red area with trigger
9. ExitPortal.prefab
    - Components: ExitPortal
    - Visual: Purple cylinder portal
10. PauseMenu.prefab
    - Complete pause menu UI with buttons

SCENES (Assets/Scenes/)

1. MainMenu.unity
    - Main menu UI with title, buttons, language selector
    - GameManager and AudioManager
2. Level1_Tutorial.unity
    - Basic tutorial level with 3 crystals
    - Platforms for gravity practice
    - Exit portal
3. Level2_Platforms.unity
    - Moving platforms challenge
    - 5 crystals scattered across platforms
    - Larger play area (30x30)
4. Level3_Hazards.unity
    - Hazard zones to avoid
    - Safe platforms between hazards

    - 4 crystals + checkpoint

5. Level4_Mechanisms.unity

    - Energy barriers and pressure plates
    - 6 crystals
    - Puzzle-solving elements

6. Level5_Final.unity

    - All mechanics combined
    - 3 enemies with AI
    - 10 crystals
    - 2 checkpoints
    - Largest play area (50x50)

UI ELEMENTS

Main Menu:

- Title: “GRAVITY SHIFT BATTLE”
- Subtitle
- Play, Options, Quit buttons
- Language selector (English, 中文, 日本語, 한국어)

In-Game HUD:

- Energy bar (top-left)
- Crystal counter (top-left)
- Score display (top-left)
- Gravity direction indicator (top-center)

Pause Menu:

- Resume, Restart, Main Menu, Quit buttons
- Semi-transparent overlay

==

1. TECHNICAL IMPLEMENTATION DETAILS
   ==

MATERIAL CREATION

- Uses Unity’s Standard Shader
- Configures rendering modes (Opaque/Transparent)
- Sets blend modes for transparency
- Applies color properties via shader parameters

PREFAB GENERATION

- Creates GameObjects programmatically
- Adds primitive shapes for visual representation
- Attaches MonoBehaviour scripts via reflection
- Configures component properties in Inspector
- Saves as prefab assets using PrefabUtility API

SCENE POPULATION

- Opens scenes via EditorSceneManager
- Instantiates prefabs using PrefabUtility
- Creates level geometry with primitive cubes
- Positions objects based on level design
- Marks scenes as dirty and saves changes

UI CREATION

- Creates Canvas with appropriate render mode
- Generates UI elements (Text, Image, Button)
- Configures RectTransform for layout
- Sets up button click handlers
- Applies color schemes and fonts

==

1. TROUBLESHOOTING
   ==

ISSUE 1: “Script not found” errors

CAUSE: C# scripts not compiled yet SOLUTION: Wait for Unity to finish compiling all scripts before running tool

ISSUE 2: Materials appear pink/magenta

CAUSE: Shader not found or material not applied SOLUTION: Re-run “1. Create Materials” module

ISSUE 3: Prefabs missing components

CAUSE: Scripts not attached correctly SOLUTION: Check Console for errors, ensure all C# scripts exist

ISSUE 4: Scenes appear empty

CAUSE: Prefabs not created before scene setup SOLUTION: Run complete setup or create prefabs first

ISSUE 5: UI not visible

CAUSE: Canvas not configured correctly SOLUTION: Re-run “4. Create UI Elements” module

==

1. DEVELOPMENT WORKFLOW INTEGRATION
   ==

RECOMMENDED WORKFLOW:

1. Write/modify C# scripts
2. Test individual scripts in simple scenes
3. Run automated setup tool to regenerate assets
4. Test complete game functionality
5. Iterate and refine

WHEN TO RE-RUN THE TOOL:

- After adding new scripts
- After modifying component requirements

- After changing material properties
- When setting up project on a new machine
- When resetting project to clean state

BACKUP STRATEGY:

- Tool can be re-run at any time without data loss
- Existing assets are updated, not duplicated
- Use version control (Git) for code backup
- Unity scenes are saved automatically

==

1. EDUCATIONAL VALUE
   ==

LEARNING OUTCOMES:

This tool demonstrates:

1. Unity Editor scripting and automation
2. Programmatic asset creation
3. Component-based architecture
4. Scene management via code
5. UI generation and layout
6. Material and shader configuration
7. Prefab workflow and instantiation
8. Software engineering best practices

SKILLS DEVELOPED:

- C# programming for Unity Editor
- Unity API usage (EditorWindow, AssetDatabase, PrefabUtility)
- Tool development for game production
- Workflow optimization
- Code organization and modularity

==

1. CONCLUSION
   ==

The Automated Project Setup Tool represents a professional approach to game development, emphasizing:

- EFFICIENCY: Automated tasks save development time
- QUALITY: Consistent setup reduces bugs
- SCALABILITY: Easy to extend with new features
- MAINTAINABILITY: Clear code structure and documentation
- PROFESSIONALISM: Industry-standard practices

This tool transforms the project setup from a manual, error-prone process into a reliable, repeatable, and documented workflow.

==

END OF GUIDE