==========

UNITY PROJECT SETUP GUIDE - GRAVITY SHIFT

This guide explains how to complete the Unity project setup after cloning from GitHub, including creating scenes, prefabs, and assigning references.

==========

STEP 1: CLONE AND OPEN PROJECT

1. Clone GitHub Repository: git clone https://github.com/Xiangfeng-Ding/GravityShift.git cd GravityShift
2. Open in Unity Hub:
    - Open Unity Hub
    - Click “Add” > Select GravityShift folder
    - Ensure Unity 2022.3 LTS is selected
    - Click on project to open
3. Wait for Import:
    - Unity will import all scripts and settings
    - This may take 2-5 minutes on first open

==========

STEP 2: CREATE PLAYER PREFAB

1. Create Player GameObject:
    - Hierarchy > Right-click > 3D Object > Capsule
    - Rename to “Player”

- Tag: Player
- Layer: Player

2. Add Components:
   - Add Component > Character Controller
     - Height: 2
     - Radius: 0.5
     - Center: (0, 1, 0)
   - Add Component > PlayerController script
   - Add Component > GravityController script
   - Add Component > PlayerEnergy script
3. Create Camera Holder:
   - Right-click Player > Create Empty
   - Rename to “CameraHolder”
   - Position: (0, 0.6, 0)
   - Right-click CameraHolder > Camera
   - Rename to “PlayerCamera”
   - Tag: MainCamera
   - Position: (0, 0, 0)
4. Assign References in Inspector: PlayerController:
   - Camera Transform: Drag PlayerCamera
   - Ground Layer: Select “Ground”

GravityController:

- Camera Transform: Drag PlayerCamera
- Player Model: Drag Player (or create a child visual object)

1. Save as Prefab:
   - Drag Player from Hierarchy to Assets/Prefabs/
   - Name: “Player.prefab”

==========

STEP 3: CREATE CRYSTAL PREFAB

1. Create Crystal GameObject:
    - Hierarchy > 3D Object > Sphere
    - Rename to “Crystal”
    - Tag: Crystal
    - Layer: Crystal
    - Scale: (0.5, 0.5, 0.5)
2. Add Components:
    - Add Component > Sphere Collider
        - Is Trigger: checked
    - Add Component > CrystalPickup script
3. Create Visual (Optional):
    - Add Material with emissive cyan color
    - Add Point Light (Color: cyan, Range: 3)
4. Save as Prefab:
    - Drag to Assets/Prefabs/Crystal.prefab

==========

STEP 4: CREATE CHECKPOINT PREFAB

1. Create Checkpoint GameObject:
    - Hierarchy > 3D Object > Cylinder
    - Rename to “Checkpoint”
    - Tag: Checkpoint
    - Scale: (2, 0.2, 2)

2. Add Components:
    - Add Component > Box Collider
        - Is Trigger: checked
        - Size: (2, 2, 2)
        - Center: (0, 1, 0)
    - Add Component > Checkpoint script
3. Create Visual Platform:
    - Create Material (green emissive for inactive, blue for active)
    - Assign to Checkpoint Renderer in script
4. Save as Prefab:
    - Drag to Assets/Prefabs/Checkpoint.prefab

==========

STEP 5: CREATE ENEMY PREFAB

1. Create Enemy GameObject:
    - Hierarchy > 3D Object > Capsule
    - Rename to “PatrolDrone”
    - Tag: Enemy
    - Layer: Enemy
    - Scale: (0.8, 0.8, 0.8)
2. Add Components:
    - Add Component > Character Controller
        - Height: 2
        - Radius: 0.5
    - Add Component > EnemyAI script
3. Create Patrol Waypoints:
    - Hierarchy > Create Empty > “PatrolRoute”

- Create 3-4 Empty children named “Waypoint1”, “Waypoint2”, etc.
- Position waypoints in a path

4. Assign References: EnemyAI:
    - Patrol Waypoints: Drag waypoint transforms
    - Detection Layers: Select “Player”
    - Obstacle Layers: Select “Ground”
5. Save as Prefab:
    - Drag to Assets/Prefabs/Enemy.prefab

==========

STEP 6: CREATE PLATFORM PREFABS

1. Static Platform:
    - Hierarchy > 3D Object > Cube
    - Rename to “Platform”
    - Tag: Ground
    - Layer: Ground
    - Scale: (5, 0.5, 5)
    - Save as Prefab
2. Moving Platform:
    - Create Cube > “MovingPlatform”
    - Add Component > Box Collider (Is Trigger: checked)
    - Add Component > MovingPlatform script
    - Create Empty waypoint objects
    - Assign waypoints in script
    - Save as Prefab

==========

STEP 7: CREATE HAZARD PREFABS

1. Death Zone:
    - Hierarchy > 3D Object > Cube
    - Rename to “DeathZone”
    - Tag: Hazard
    - Layer: Hazard
    - Scale: (10, 1, 10)
    - Add Component > Box Collider (Is Trigger: checked)
    - Add Component > HazardZone script
    - Material: Red transparent
    - Save as Prefab
2. Energy Barrier:
    - Create Cube > “EnergyBarrier”
    - Add Component > Box Collider
    - Add Component > EnergyBarrier script
    - Material: Red emissive transparent
    - Save as Prefab

==========

STEP 8: CREATE UI CANVAS

1. Create Main Canvas:
    - Hierarchy > UI > Canvas
    - Rename to “GameUI”
    - Render Mode: Screen Space - Overlay
    - Add Component > UIManager script
2. Create HUD Panel:
    - Right-click Canvas > UI > Panel

- Rename to “HUD”
- Anchor: Top stretch

Create children:

- Energy Bar (UI > Slider)
- Crystal Text (UI > TextMeshPro)
- Timer Text (UI > TextMeshPro)
- Gravity Direction Text (UI > TextMeshPro)

1. Create Pause Menu Panel:
   - Right-click Canvas > UI > Panel
   - Rename to “PauseMenu”
   - Set Active: false

Create buttons:

- Resume Button
- Restart Button
- Main Menu Button

1. Create End Level Panel:
   - Similar structure with score display
   - Set Active: false
2. Assign References:
   - In UIManager script, drag all UI elements to corresponding fields

==========

STEP 9: CREATE MANAGERS GAMEOBJECT

1. Create Managers:
   - Hierarchy > Create Empty > “GameManager”
   - Add Component > GameManager script
   - Create Empty > “LanguageManager”

- Add Component > LanguageManager script
- Create Empty > “AudioManager”
- Add Component > AudioManager script

2. Assign References: GameManager:
   - Player: Drag Player from scene
   - Total Crystals: Will auto-detect

AudioManager:

- Create child AudioSource objects for Music and SFX
- Assign audio clips (when available)

==========

STEP 10: BUILD LEVEL 1 - TUTORIAL

1. Create New Scene:
   - File > New Scene
   - Save as “Level1_Tutorial” in Assets/Scenes/
2. Add Essential Objects:
   - Directional Light
   - Drag Player prefab into scene
   - Position at (0, 2, 0)
3. Build Environment:
   - Create floor platform (10x1x10 cube, Layer: Ground)
   - Create walls for boundaries
   - Place 5 Crystal prefabs around level
   - Place 2 Checkpoint prefabs
   - Create Exit Portal (Cylinder with ExitPortal script)
4. Add Managers:
   - Drag GameManager prefab

- Drag LanguageManager prefab
- Drag AudioManager prefab
- Drag GameUI Canvas

5. Test Scene:
   - Press Play
   - Verify player movement
   - Test gravity switching (Hold G + Arrow keys)
   - Collect crystals
   - Reach exit portal

==========

STEP 11: CREATE REMAINING LEVELS

For each level (Level2-Level5):

1. Duplicate Level1_Tutorial scene
2. Rename appropriately
3. Modify layout and difficulty:
   - Level 2: Add moving platforms, 1 enemy
   - Level 3: Add hazard zones, 2 enemies
   - Level 4: Add pressure plates, energy barriers
   - Level 5: Combine all mechanics, increase difficulty
4. Adjust GameManager settings:
   - Crystal requirements
   - Time limits
   - Checkpoint count

==========

STEP 12: CREATE MAIN MENU SCENE

1. Open MainMenu Scene:
    - Already exists in Assets/Scenes/MainMenu.unity
2. Create UI:
    - Canvas with MainMenu script
    - Title Text: “Gravity Shift”
    - Start Button
    - Settings Button
    - Quit Button
    - Language Dropdown (top-right)
3. Create Difficulty Panel:
    - Easy Button
    - Normal Button
    - Hard Button
4. Create Settings Panel:
    - Language Dropdown
    - Volume Sliders
    - Camera Sensitivity Slider
5. Assign Button Events:
    - Start Button > MainMenu.ShowDifficultyPanel()
    - Easy Button > MainMenu.StartGameEasy()
    - Normal Button > MainMenu.StartGameNormal()
    - Hard Button > MainMenu.StartGameHard()
    - Quit Button > MainMenu.QuitGame()
6. Add Managers:
    - LanguageManager
    - AudioManager

==========

STEP 13: CONFIGURE BUILD SETTINGS

1. Open Build Settings:
    - File > Build Settings
2. Verify Scene List:
    - MainMenu (index 0)
    - Level1_Tutorial (index 1)
    - Level2_Platforms (index 2)
    - Level3_Hazards (index 3)
    - Level4_Mechanisms (index 4)
    - Level5_Final (index 5)
3. Platform Settings:
    - Platform: PC, Mac & Linux Standalone
    - Architecture: x86_64
    - Compression: Default
4. Player Settings:
    - Company Name: StudentProject
    - Product Name: GravityShift
    - Default Icon: (Optional)
    - Fullscreen Mode: Fullscreen Window
    - Default Screen Width: 1920
    - Default Screen Height: 1080

==========

STEP 14: TESTING CHECKLIST

Test in Editor: [ ] Player movement works in all directions [ ] Gravity switching works (Hold G + Arrows) [ ] Energy bar depletes and regenerates [ ] Crystals can be collected [ ] Checkpoints save position [ ] Death respawns at checkpoint [ ] Enemies patrol and chase [ ] Enemies attack when in range [ ] Moving platforms work [ ] Pressure plates activate barriers [ ] Energy barriers unlock

with crystals [ ] Exit portal checks crystal requirement [ ] UI updates correctly [ ] Language switching works [ ] Pause menu works [ ] Timer counts down [ ] Score calculates correctly

Test Build: [ ] Build completes without errors [ ] Game launches successfully [ ] All scenes load correctly [ ] Performance is acceptable (30+ FPS) [ ] No critical bugs

==========

STEP 15: FINAL POLISH (OPTIONAL)

Visual Enhancements:

- Add materials and textures
- Create particle effects for crystals
- Add lighting and shadows
- Create skybox
- Add post-processing effects

Audio:

- Add background music
- Add sound effects for:
  - Gravity switch
  - Crystal pickup
  - Checkpoint activation
  - Enemy alert
  - Player death
  - Level complete
  - UI button clicks

Gameplay Polish:

- Add tutorial text prompts
- Create level intro/outro sequences
- Add camera shake effects
- Improve enemy visual feedback
- Add minimap (optional)

==========

TROUBLESHOOTING

Issue: Scripts show “Missing Reference” Solution: Assign references in Inspector manually

Issue: Player falls through floor Solution: Ensure floor has Collider and Layer is “Ground”

Issue: Gravity doesn’t change Solution: Check GravityController is attached and energy is available

Issue: UI doesn’t update Solution: Verify UIManager references are assigned

Issue: Build fails Solution: Check Console for errors, ensure all scenes are in Build Settings

Issue: Enemy doesn’t move Solution: Assign patrol waypoints in EnemyAI script

Issue: Language doesn’t switch Solution: Ensure LanguageManager exists in scene

==========

NEXT STEPS

After completing Unity setup:

1. Test all levels thoroughly
2. Record 5-minute video demo
3. Write game design report (8 pages max)
4. Prepare classroom presentation
5. Submit to Blackboard before deadline (04/Mar/2026 15:00)

Good luck with your project!

==========

END OF SETUP GUIDE