

PREPARE

The included weapon mesh is a placeholder. Replace it with your own visual when you are ready. A final art model and a complete sound library are not included.

This package does not require URP. It does not require Shader Graph. It does not require third-party assets.

If a clean Unity 6 project does not already contain UGUI, open **Window > Package Management > Package Manager**, install **Unity UI (UGUI)**, and wait for compilation before using the **Wizard**.

For UNITY 6xxx

1. Open **Edit > Project Settings**
2. Select **Player**
3. Open **Other Settings**
4. Find **Active Input Handling**
5. Select **Both**
6. Allow Unity to restart the Editor if requested

PLAY DEMO

1. Select **Tools > Gravity Gun System > Add Demo Scenes to Build Settings**
2. Choose **Add After Existing** to preserve user scenes and append all five demos.
3. Choose **Replace Build Settings** only when you intentionally want the five demos to replace the current list.
4. Choose **Cancel** to make no change.
5. Open the **Launcher scene** and press **Play**
6. Select **FPS Mobile, FPS Desktop, TPS Mobile or TPS Desktop**

QUICK START

1. In the Unity top menu, select **Tools > Gravity Gun System > Setup Wizard**

The Wizard contains:

- **Setup Source** — Existing Player integration or Playable Starter.

Existing Player adds the Gravity Gun, input integration, UI and an EventSystem when required. It does not create player movement or camera control.

Playable Starter creates a controllable player, camera controller, Gravity Gun, matching input, UI, EventSystem and starter environment.

- **Preset** — FPS Desktop, FPS Mobile, TPS Desktop or TPS Mobile
- **Player Root** — the top-level object owned by your movement controller
- **Aim Camera** — the Camera used for the center-screen grab ray
- **Weapon Mount** — the Transform that parents the weapon
- **Refresh Scene References** — recovers valid generated or assigned references
- **Create GravityGunWeaponMount** — creates a mount under the expected FPS or TPS parent
- **Install / Repair Setup** — installs or repairs the selected setup

- **Remove Installed Gravity Gun Setup** — removes Wizard-owned objects

The Gravity Gun Inspector also has **Validate Setup**. The Wizard displays validation results after installation.

Integrate With an Existing FPS Player

Player Root is the top-level object containing your movement controller. **Aim Camera** is the Camera used to aim. **Weapon Mount** is the parent for the Gravity Gun visual and muzzle.

1. Select your player in the Hierarchy
2. Find the **Camera** used by the player
3. Right-click the **Camera** and select **Create Empty**
4. Rename the child to **GravityGunWeaponMount**
5. Reset the mount Transform to local **position** (0, 0, 0), **rotation** (0, 0, 0) and **scale** (1, 1, 1)
6. Open the Setup Wizard
7. Select **Integrate With Existing Player**.
8. Select **FPS Desktop** or **FPS Mobile**
9. Drag the player root into **Player Root**
10. Drag the player's Camera component into **Aim Camera**
11. Drag **GravityGunWeaponMount** into **Weapon Mount**
12. Click **Install / Repair Setup**
13. Press **Play**
14. Move the **Weapon Mount** until the placeholder weapon fits your view.

Integrate With an Existing TPS Player

A typical TPS hierarchy is:

```
Player Root
├── Camera Rig or Aim Camera
├── GravityGunWeaponMount
│   └── GravityGunSystem
```

The **Weapon Mount** can be under **Player Root**, a hand bone or an existing weapon socket. It should not automatically be placed under the **Camera**.

1. Select the player root
2. Create an **empty child**
3. Rename it to **GravityGunWeaponMount**
4. Move it to the player's hand or weapon socket
5. Open the **Setup Wizard**
6. Select **Integrate With Existing Player**
7. Select **TPS Desktop** or **TPS Mobile**
8. Assign **Player Root**
9. Assign the **Camera** used by the crosshair to **Aim Camera**
10. Assign the hand/socket **Transform** to **Weapon Mount**
11. Click **Install / Repair Setup**
12. Press **Play**
13. Adjust the **Weapon Mount** position and rotation for your character

Replacing the Placeholder Weapon Visual

1. In the Project window, open **Prefabs/Core/GravityGunSystem.prefab** in Prefab Mode.
2. Locate the placeholder visual below the prefab root
3. Keep the root components unchanged
4. Keep the **Muzzle** object and its reference
5. Replace only the visual mesh/model
6. Position the new model around the existing **Muzzle**.
7. Save the prefab

Do Not Delete: **GravityGun**, **GravityGunInputAdapter**, **GravityGunEquipment**, **LineRenderer**, **Muzzle**, highlighter or required module components.

WORK WITH OBJECTS

1. Create grab object

1. Create or select a **GameObject**
2. Add a **Collider** that matches its shape
3. Add a **Rigidbody**
4. Disable **Is Kinematic**
5. Put the object on a layer included in the **Gravity Gun Grab Mask**
6. Place it within **Max Distance**
7. Press **Play**, aim at the object and use **Grab**

Rigidbody mass affects movement response. The Collider provides ray and collision geometry. The Rigidbody must be on the same object hit by the ray, or on the collider's Rigidbody root.

Grab Mask and Layers

1. Select the installed **GravityGunSystem** object
2. Find the **GravityGun** component
3. Find **Grab Mask**
4. Open the layer dropdown
5. Select the layers containing grabbable objects
6. Exclude the player and weapon layers when necessary

If an outline appears but **Grab** fails, check that the outlined **Collider** resolves to a non-kinematic **Rigidbody** and that its layer is included in **Grab Mask**

2. Duplicate is opt-in.

1. Select an object that may be duplicated
2. Click **Add Component**
3. Add **GravityGunDuplicable**
4. Enable **Allow Duplication**
5. Set **Duplicate Root** if the whole hierarchy should be copied from another root.
6. Set **Maximum Active Copies**.
7. Configure **Spawn Offset** if required.
8. Press **Play** and grab the object
9. Press G on Desktop or use **DUPLICATE** on Mobile

The clone is created once and automatically becomes the held object. Objects without GravityGunDuplicable are rejected

3. Using Multi-Grab

The default maximum is five objects. One beam points to the group center. Distance, rotation, throw, freeze and release operate on the group.

1. Press **M** on Desktop or use **MULTI** on Mobile.
2. Aim at the first **Rigidbody** and use **Grab** to add it
3. Repeat for up to five compatible objects
4. **Grab** a current member again to remove it
5. Change distance or enter Rotate Mode to manipulate the group
6. **Throw, freeze or release** the group

Desktop uses the normal **Grab/Throw/Freeze/Distance/Rotate** controls while **Multi-Grab** is active. Mobile uses the same visible action buttons.

The production beam is procedural and does not use a texture.

4. Outline Settings

The Built-in highlighter supports Hovered, Grabbed and multiple grabbed targets. It has no Quick Outline dependency.

1. Select **GravityGunSystem**
2. Find **BuiltInGravityGunHighlighter**
3. Set **Hover Color, Grabbed Color, Outline Width** and **Outline Mode**
4. To customize one target, add **GravityGunOutlineTarget** to that object
5. Use its visual root, renderer exclusion, width multiplier and color overrides as needed

Disable the highlighter component if no outline is wanted. Do not replace its materials while targets are highlighted.

Beam Settings

1. Select GravityGunSystem .
2. Find the Beam fields on GravityGun .
3. Configure Beam Enabled , Beam Color , Beam Core Color , Beam Start Width , Beam End Width , Beam Intensity , Beam Edge Softness , Beam Pulse Speed , Beam Pulse Frequency and Beam Pulse Strength .
4. Set Pulse Strength to 0 for a stable beam.
5. Disable Beam Enabled to run without a beam.

Six Complete Scenarios

Scenario 1 — Create FPS Desktop Starter

Starting condition: an empty scene and no player controller.

1. Open the **Wizard**.
2. Choose **Create Playable Starter** and **FPS Desktop**
3. Click **Install / Repair Setup**
4. Save and press **Play**

Scenario 2 — Create FPS Mobile Starter

Starting condition: an empty landscape-oriented scene

1. Choose **Create Playable Starter** and **FPS Mobile**
2. Install, save and press **Play**

Scenario 3 — Create TPS Desktop Starter

Starting condition: an empty scene

1. Choose **Create Playable Starter** and **TPS Desktop**.
2. Install, save and press **Play**.

Scenario 4 — Create TPS Mobile Starter

Starting condition: an empty landscape-oriented scene

1. Choose **Create Playable Starter** and **TPS Mobile**
2. Install, save and press **Play**

Scenario 5 — Integrate With Existing FPS Player

Starting condition: a working FPS Player Root and Camera.

1. Create **GravityGunWeaponMount** below the **Camera**
2. Choose **Integrate With Existing Player** and an **FPS preset**
3. Assign **Player Root**, **Aim Camera** and **Weapon Mount**
4. Install and press **Play**

Scenario 6 — Integrate With Existing TPS Player

Starting condition: a working TPS Player Root, camera and hand/socket location

1. Create or select a **Weapon Mount** below **Player Root**/hand/socket
2. Choose **Integrate With Existing Player** and a **TPS preset**
3. Assign **Player Root**, independent **Aim Camera** and **Weapon Mount**
4. Install and adjust the mount

Gravity Gun System Customization Reference

1. Gravity Gun Customization Overview

The primary settings are located on the **GravityGun** component of the main **GravityGunSystem** object. These settings control:

- target selection and physics interaction;
- initial and adjustable hold distance;
- hold response and throw speed;
- object rotation and freezing;
- optional duplication and Multi-Grab modules;
- built-in or custom beam visuals;
- built-in or custom target highlighting;
- action audio;
- action effects.

Beam and built-in outline settings support live updates while the Editor is in Play Mode.

2. How to Open the Settings

1. Find the installed **Gravity Gun System** instance in the Hierarchy.
2. Select the object that contains the **GravityGun** component.
3. Locate the required section in the Inspector, such as **Audio**, **Effects**, **Beam Visual**, **Settings**, or **Distance** .

3. References, Targeting, Physics, and Distance

References

Field	Purpose	Effect of the setting
Player Camera	Camera used to cast the center-screen targeting ray and determine forward hold and throw directions	Assign the gameplay aim camera. Without it, camera-based Grab and Throw cannot operate correctly
Line Renderer	Renderer used by the built-in procedural beam	Required for Built-In Procedural beam mode; it does not control physics
Muzzle	World-space start point of the beam and preferred spawn point of Throw Effect	Moving it changes the visible beam origin and Throw Effect origin, not the targeting ray
Highlighter Behaviour	A MonoBehaviour implementing the Gravity Gun highlighting interface	Assign the built-in highlighter or a compatible custom implementation. Empty is valid but provides no outline
Duplicate Module	Optional GravityGunDuplicateModule	Enables Duplicate commands when the held target is configured with GravityGunDuplicable
Multi Grab Module	Optional GravityGunMultiGrabModule	Enables Multi-Grab mode and group formation controls

Settings

Field	Current Core default	Behaviour
Grab Mask	Project-specific layer mask	Only colliders on included layers can be targeted. The raycast must resolve a Rigidbody
Max Distance	15	Maximum camera-to-target distance used by the targeting ray and Multi-Grab validation
Hold Force	15	Controls how quickly held Rigidbody centers move toward their calculated hold positions. Higher values respond more aggressively
Throw Force	20	Sets the forward velocity used by Single Throw and Multi Throw

Targeting uses a ray from the center of Player Camera . A collider hit is accepted only when it is inside Grab Mask , within Max Distance, and resolves to a Rigidbody.

Distance

Field	Current Core default	Behaviour
Min Grab Distance	2	Minimum allowed camera-to-hold distance. The initial distance and Distance Minus command are clamped to this value
Max Grab Distance Limit	25	Maximum allowed hold distance. It can be greater than the initial targeting Max Distance
Distance Step	0,5	Amount added or removed by each Distance Plus/Minus command. Applies to both Single Grab and Multi-Grab

The initial hold distance is calculated from the camera position to Rigidbody.worldCenterOfMass , then clamped between Min Grab Distance and Max Grab Distance Limit. Distance adjustment changes the hold distance only; it does not change the targeting ray range.

Rotation and Freeze

Rotation commands operate on the currently held Rigidbody or the active **Multi-Grab** group. Desktop and Mobile demo input components currently use a **Rotation Step** of 15 degrees. The Rotate90 command applies a 90-degree yaw step.

Freeze requires a **Freezable** component on the held object. The normal successful **Grab** path unfreezes an already frozen target before holding it. Freezing releases the object after its frozen state is applied.

4. Audio Settings

All action clips are played through the assigned Audio Source by using AudioSource.PlayOneShot . Clips can overlap. Leaving an individual clip empty does not block its action; the action simply runs without that sound.

The Core prefab contains an AudioSource configured as 2D audio by default. To make action sounds spatial, change the **AudioSource Spatial Blend** toward 3D and configure its normal Unity

Field	Trigger	Single grab	Multi-Grab
Audio Source	Used by every action clip	Plays Grab Clip	Plays the applicable group or per-object clips
Grab Clip	Successful Grab	Plays once after a Rigidbody is successfully grabbed	Plays once for every newly and successfully added Rigidbody
Drop Clip	Release	Plays once for normal Release	Plays for an individual removal; group Release plays one shared sound
Throw Clip	Release	Plays once for normal Release	Plays for an individual removal; group Release plays one shared sound
Freeze Clip	Throw	Plays once	Plays once for the whole group
Rotate Clip	A rotation step	Plays when a rotation command applies a step of at least 10 degrees; Rotate90 always plays it	Same group-step behaviour

A successful Duplicate currently uses the normal Drop and Grab paths, so it can play Drop Clip followed by Grab Clip. This is the current behaviour rather than a separate Duplicate audio hook.

5. Action Effects

The `Effects` section contains:

- Grab Effect
- Throw Effect
- Freeze Effect
- Effect Lifetime

Each effect field accepts a **GameObject**. Assigning a prefab asset from the **Project** window is recommended. An effect prefab may contain:

- a Particle System
- a MeshRenderer
- a Light
- an Animator
- a custom MonoBehaviour
- any combination of these components

For Particle System prefabs, enable Play On Awake or include a script that starts the effect. Gravity Gun does not call `ParticleSystem.Play()` manually. A looping Particle System is technically supported, but its instantiated `GameObject` is still destroyed after Effect Lifetime.

Grab Effect

- Created only after a successful **Grab**.
- Single Grab creates one instance at `Rigidbody.worldCenterOfMass`.
- **Multi-Grab** creates one instance at the center of every newly and successfully added Rigidbody.
- **Duplicate** and automatic clone Grab create one instance on the clone through the normal successful **Grab** path
- Hover does not create the effect
- A failed raycast, invalid target, rejected Multi-Grab target, or other failed Grab does not create it
- Attempting to grab a Rigidbody that is already held does not duplicate the effect

Throw Effect

- Single Throw creates one effect instance.
- Its preferred position is Muzzle
- If Muzzle is not assigned, Single Throw falls back to `thrownBody.worldCenterOfMass`
- Multi Throw creates one shared effect, not one per Rigidbody
- Its preferred position is Muzzle.
- Without Muzzle, Multi Throw uses the valid group center calculated before the Multi-Grab list is cleared.
- The effect is created as part of the Throw action after throw state has been applied.

Freeze Effect

- Single Freeze creates one effect at `frozenBody.worldCenterOfMass`.
- Multi Freeze creates one shared effect at the valid center of the group.
- Multi Freeze does not create a separate instance for every Rigidbody.
- The effect is created when the freeze action is applied and the held state is cleared.

Effect Lifetime

Effect Lifetime is the time in seconds before an instantiated action-effect GameObject is automatically destroyed.

- Core prefab default: 2 seconds.
- Inspector minimum: 0.05 seconds.
- Runtime cleanup also clamps the effective value to at least 0.05 seconds.
- The same lifetime is used by Grab Effect, Throw Effect, and Freeze Effect.

Technical Behaviour

Field	Trigger	Spawn Position	Multi-Grab Behaviour
Grab Effect	Successful Grab	Rigidbody worldCenterOfMass	One per successfully added Rigidbody
Throw Effect	Throw	Muzzle; body/group-center fallback	One shared effect for group Throw
Freeze Effect	Freeze	Single body center or Multi group center	One shared effect for group Freeze
Effect Lifetime	Applies when any action effect is spawned	Applies when any action effect is spawned	

Null effect fields and invalid world positions are skipped safely.

6. Beam Visual Settings

Beam rendering is separated from Gravity Gun targeting and physics. A beam provider receives visual state after the core has determined the current Muzzle and held-object/group positions. It does not perform Grab, Release, Throw, Freeze, or raycasts.

Beam Visual Mode

Disabled

- No beam is displayed.
- All Gravity Gun mechanics continue to work.
- A built-in material and custom provider are not required in this mode.

Built-In Procedural

- Uses the included GravityGunProceduralBeamVisual provider and assigned Line Renderer.
- Supports **Straight** and **Curved** shapes.
- Uses preallocated point storage; the point array changes only when the required point count changes.
- Beam fields can be changed in **Play Mode** and are refreshed immediately by the custom Inspector.

Custom

- Uses the MonoBehaviour assigned to Custom Beam Visual
- The assigned component must implement IGravityGunBeamVisual
- The previously active built-in visual is hidden and disposed when the provider changes
- Physics and held-object state remain unchanged

Custom Beam Visual

1. Create a MonoBehaviour.
2. Implement IGravityGunBeamVisual .
3. Add the component to a GameObject in the installed setup or custom prefab.
4. Assign that MonoBehaviour to Custom Beam Visual .
5. Select Beam Visual Mode = Custom .

Current interface:

```
public interface IGravityGunBeamVisual
{
    void Initialize();
    void Show();
    void Hide();
    void UpdateVisual(in GravityGunBeamVisualState state);
    void Dispose();
}
```

GravityGunBeamVisualState supplies:

- **Start** - World-space Muzzle/beam origin
- **End** - World-space held Rigidbody center or Multi-Grab group center
- **Color** - Current action color selected by the core
- **Is Active** - Whether the state represents an active beam update
- **Is Multi Grab** - Whether Multi-Grab mode is active
- **Held Object Count** - Number of held objects represented by the beam

A custom provider should cache its component references during initialization, avoid per-frame allocations, hide only its own visual, and release only resources that it created in Dispose() .

Built-In Procedural Fields

- **Procedural Beam Visual** - Responsible for configuring and updating the built-in Line Renderer
- **Beam Material**
- **Grab Beam Color**
- **Freeze Beam Color**
- **Beam Color**
- **Beam Core Color**
- **Beam Start Width**
- **Beam End Width** - Width at the held-object endpoint
- **Beam Intensity** - Shader brightness/intensity parameter
- **Beam Edge Softness** - Controls edge falloff in the included beam shader
- **Beam Pulse Speed**

- **Beam Pulse Frequency** - Number/frequency of pulse repetitions along the visual
- **Beam Pulse Strength** - Strength of the pulse modulation
- **Beam Shape** - Selects a two-point straight beam or the procedural curved path

Curved Beam Fields

- **Segments** - Number of LineRenderer points
- **Curve Down** - Adds downward sag to the Bezier control point. Negative values bend upward
- **Curve Side** - Adds a sideways offset relative to the beam direction. Negative values reverse the side
- **Curve Strength By Distance** - Scales the curve strength with beam length, capped by the provider's distance scaling. |
- **Wave Strength** - Amplitude of the sideways animated wave. Set to 0 to disable wave displacement
- **Wave Frequency** - Number/frequency of wave cycles along the beam. Minimum is 0
- **Wave Speed** - Time-based motion speed. Negative values reverse animation direction

The first and last procedural points are always overwritten with the exact Start and End state positions. Short beams automatically reduce curvature and wave displacement

7. Outline and Highlight Settings

Outline changes refresh active highlights during Play Mode

Materials

- **Outline Mask Material** - Mask pass used by the built-in outline. Required when Outline Enabled is active
- **Outline Fill Material** - Fill pass used to draw the expanded colored outline. Required when Outline Enabled is active

Appearance

- **Outline Enabled** - Globally enables or disables built-in outline drawing without disabling targeting
- **Hover Color** - Color used for a valid target under the center-screen ray
- **Grabbed Color** - Color used by a Single Grab target
- **Multi Grab Color** - Color used by members of the active Multi-Grab group
- **Outline Width** - Base expansion width
- **Occluded Color** - Color of the occluded pass in Visible And Occluded mode
- **Occluded Alpha** - Opacity of that occluded pass
- **Outline Mode** - Visible And Occluded | Selects how depth testing is applied

Outline Mode values:

- **Visible Only:** draws the outline only where the highlighted object is visible through normal depth testing.
- **Always Visible:** draws the main outline regardless of occlusion.
- **Visible And Occluded:** draws a normal visible outline plus an occluded pass using Occluded Color and Occluded Alpha .

Renderer Collection

- **Include Child Renderers** = Collects renderers under the target visual root instead of only renderers on that root
- **Include Inactive Renderers** - Includes inactive child renderer components during collection when child collection is enabled
- **Ignore Particle Renderers** - Excludes ParticleSystemRenderer components from highlighting
- **Ignore Trail Renderers** - Excludes TrailRenderer components
- **Ignore Line Renderers** - Excludes LineRenderer components, which prevents the beam or target lines from receiving an outline

The built-in implementation temporarily appends its mask/fill materials and restores the original renderer materials and MaterialPropertyBlock state when a highlight is removed.

Per-Target GravityGunOutlineTarget

Add GravityGunOutlineTarget to a target or a parent to override outline collection and appearance for that target.

- **Outline Enabled** - Allows this target to opt out without disabling the global highlighter
- **Visual Root** - Root from which renderers are collected. Empty uses the component's Transform
- **Override Include Child Renderers** - Enables the per-target child-renderer setting
- **Include Child Renderers** - Per-target collection choice when the override is enabled
- **Width Multiplier** - Multiplies the global Outline Width for this target; minimum is 0
- **Override Hover Color** - Enables the target-specific Hover Color
- **Hover Color** - Hover color used only when its override is enabled
- **Override Grabbed Color** - Enables the target-specific Grabbed Color
- **Grabbed Color** - Used for grabbed states when its override is enable|
- **Excluded Renderers** - Explicit renderers that must not receive the outline

8. Duplicate Settings

Duplicate is optional

- **Allow Duplication** - Allows or rejects Duplicate for this definition. |
- **Duplicate Root** - Root GameObject cloned by Duplicate. Empty uses the component's Transform. The held Rigidbody must be this root or a child
- **Maximum Active Copies** - Maximum live clones tracked for this source; Inspector range 1–100. Destroyed clones stop counting
- **Copy Rigidbody Settings** - Copies mass, drag, angular drag, interpolation, collision detection, constraints, and kinematic state to the clone Rigidbody
- **Copy Runtime Transform** - Starts the clone from the source's current position, rotation, and authored scale behaviour. Disabled uses transform values recorded at Awake
- **Spawn Offset** - Offset from the source position. With an Aim Camera it is interpreted along camera right/up/forward axes
- **Preserve Parent** - Parents the clone beside the source under the same parent when enabled; otherwise the clone has no parent

Successful Duplicate releases the original and automatically grabs the clone.

9. Multi-Grab Settings

The optional GravityGunMultiGrabModule contains the group formation settings:

- **Maximum Grabbed Objects** - Maximum simultaneous group size; Inspector range 1–20
- **Formation Spacing** - Horizontal and vertical spacing between formation slots; minimum is 0.1
- **Formation Columns** - Maximum columns before additional rows are created; Inspector range 1–5

The first added object's distance establishes the group's hold distance. Group Distance Plus/Minus, rotation, release, throw, and freeze then operate on the formation as a unit.

Multi-Grab also rejects:

- inactive or invalid Rigidbody targets;
- targets outside Grab Mask or Max Distance;
- a target already in the group;
- targets that exceed Maximum Grabbed Objects;
- parent/child hierarchy conflicts between group members.