Customized Objects Gravity System - Documentation

Overview

The Customized Objects Gravity System is a dynamic gravity tool that enables game objects to interact with the world around them in an entirely new way. Instead of being limited to downward gravity, objects can now be pulled toward the nearest surface, allowing for gameplay across walls, ceilings, or irregular terrain. Gravity direction is continuously recalculated, allowing for seamless transitions as the object moves around.

Whether you're making a space exploration game, a zero-gravity puzzle, or an unconventional platformer, this system provides a foundation for unique physics-based behavior.

How It Works

This system uses a single script: customized_objects_gravity. It works by constantly checking for the closest surface near an object and calculating a gravity vector that pulls the object toward it. This is done using Unity's physics system (via raycasting and collider checks) and then applying that force through a ConstantForce component.

Here's a simplified breakdown:
- Every few frames, the script checks surrounding colliders using a sphere overlap.
- It finds the closest point on the nearest collider that belongs to a designated gravity layer.
- A raycast determines the surface normal from the object to that point.
- The gravity force is recalculated and applied using ConstantForce, pulling the object toward the surface.

This allows the object to simulate gravity that sticks to and flows along complex geometry — even upside down.

Setup Instructions

1. Add Components to Your Object
  - Add a Rigidbody.
    - Uncheck "Use Gravity".
  - Add a Constant Force component.
  - Attach the customized_objects_gravity script to the same object.

2. Layer Setup
  - Make sure your gravity ground objects (surfaces you want to attract objects to) are on a specific layer (e.g., "GravitySurface").
  - Set this layer in the script's Layer Mask field.
  - Important: The objects that use custom gravity must NOT be on this same layer — otherwise, they'll attract each other, leading to strange behavior.

3. Adjust Parameters (Optional)
  - `radius`: Controls how far around the object it searches for gravity surfaces.
  - `gravity_force`: Determines how strong the pull is. Higher = faster.
  - `mass`, `drag`, `angular drag` (Rigidbody): Can be fine-tuned for more or less resistance or weightiness.

Customization Points

You can tweak several settings to match your gameplay style:
- Gravity Range: Control how far objects can search for nearby surfaces.
- Gravity Strength: Increase or decrease the pull.
- Physics Tuning: Adjust Rigidbody values for effects like floating, slow falling, or snappy magnetism.

Troubleshooting

Object Doesn't Move or Float
- Make sure Rigidbody's Use Gravity is turned off.
- Check that Constant Force is present.
- Confirm that gravity_force is greater than zero.

Object Attracted to Itself or Behaves Weirdly
- Double-check that the object using the script is on a different layer than the gravity surfaces.
- Ensure the Layer Mask in the script does NOT include the layer of the gravity object.

Object Snaps or Twitches
- This could happen if there are very tight, thin objects or overlapping colliders.
- Try increasing the radius slightly or smoothing normals (or reduce the frequency of surface checks).

Script Reference (Summary)

// customized_objects_gravity.cs
// Finds nearest surface and applies gravity toward it.

- gravity_force: strength of gravity.
- radius: range to detect nearby surfaces.
- layer_mask: which layers are considered valid gravity surfaces.
- cf.force: updated continuously based on closest surface normal.

Final Notes

This system gives your game objects freedom to move in ways standard gravity doesn't allow. Whether it's crawling around a space station, running on planetary interiors, or gliding up walls, this tool offers creative ways to design your levels and characters.