Physics Demo readme

Thanks for downloading the official "Physics Demo" for Adventure Creator. To play it, an installation of Adventure Creator v1.69.0 or later is required.

This Unity package is for educational purposes only, and the assets within cannot be redistributed or sold without expression permission from their respective owners.

Installation

1. Extract this package into your Adventure Creator project
2. Map the “r” key to the input button “RotateMoveable” in Unity’s Input Manager
3. Navigate to Adventure Creator → **Getting started → Load Physics demo**
4. When prompted, open the Physics demo scene and run it
5. Map the "r" key to the input button "RotateMoveable" within the Input Manager.

Details

The following describes some of the more complex elements of the demo.

Lip syncing

During the opening cutscene, the main character (Brain) has two lines spoken on-screen, and each of these are animated using BlendShape lip-syncing. The lip-sync files are generated using Sapi, and mapped to the character model via the **Shapeable** component, which is attached to the **Skinned Mesh Renderer**. The Shapeable component is used to separate the model's seven BlendShapes into Expression and Phoneme groups. The shapes in the Phoneme group match the phonemes declared in the Speech Manager’s Phonemes window.

Desk overview

After the intro has played, Brain is teleported out of the way, and the chair is made invisible - both so that the desk can be viewed properly by the player.

A **GameCamera Third Person** camera is used for examining the desk as a whole. It has drag-control enabled, so that it can be manipulated with the mouse. It also has limits imposed, so that the player cannot clip through the geometry.

In this view, the physics objects in the scene are disabled - only the "zoomed out" Hotspots are enabled. These Hotspots are replaced with close-up variants when the player clicks one to zoom in. To begin with, only the "Drawers Hotspot" is enabled, and the others are enabled as the game progresses. The other Hotspots are disabled when the game starts by attaching a **RememberHotspot** script to them, and setting their initial state to Off.

Entering and exiting close-ups

When clicking one of the "zoomed out" Hotspots, the camera switches to a new camera that focuses on a specific part of the desk. Hotspots are enabled and disabled as appropriate by using **Object: Send message** Actions.

Since the act of enabling and disabling Hotspots are quite common in this scene, a generic Cutscene called "Enable and Disable" is used to perform the task. This Cutscene uses parameters, which allows different Hotspots to be manipulated each time. By calling this Cutscene with the **ActionList: Run** Action, these parameters can be set each time it is run.

During close-ups, the "Exit closeup" Menu appears in the lower-right corner. Clicking this Menu's Button zooms the camera back out to it's starting position, as well as enabling and disabling the appropriate Hotspots. When clicked, an ActionList asset called "Exit closeup" will run, but this in turn calls a scene-based Cutscene of the same name.

Depending on which part of the desk was being focused on during the closeup, different Hotspots will need to be manipulated when zooming out. Therefore, the "Exit closeup" Cutscene relies on a string Global Variable, "Close-up type", to know which part of the screen was being focused on. This Variable is set when a close-up begins.

Drawers close-up

Each drawer is a **Draggable** object, locked to a **Straight Track** that constrains it's movement to just one direction. Such constraints are made by creating upper- and lower-limit Colliders at runtime, as child objects of each Track. To position these Colliders correctly, a **Sphere Collider** is required to be part of the Draggable's base GameObject, as this is used to position the limiting colliders. This Sphere Collider is disabled, however, so that only the Mesh Colliders in the child are interactive.

The letter in the middle drawer, which is a regular Hotspot, will move realistically when the drawer that it sits in is moved. To achieve this effect, it's Box Collider has **Is Trigger**

unchecked, and has a **Rigidbody** attached. This Rigidbody component is constrained so that it can only move and rotate appropriately.

Rock close-up

The rock is a **PickUp** object, and can be picked up and rotated. The key underneath is a child of the rock, and is removed from the scene when picked up by using the **Object: Add or remove** Action.

Because the view is zoomed in quite close to the rock when it is interactive, some precautions are taken to prevent the rock from leaving the view. Firstly, a custom **Mesh Collider** is present, in the _Colliders folder, to make the rock collide appropriately with the geometry. However, it is also possible to "throw" the rock out of the scene, if the cursor is moved fast enough. In a First Person game that has an unconstrained camera, this wouldn't be a problem, but a safeguard is required here. If the rock leaves the view, it is teleported back to it's original position. This is done with the "Return rock" Trigger, which is triggered when the rock leaves it's bounding box.

Cupboard close-up

The cupboard is a **Draggable** object, locked to a **Hinge Track**. The door's position is unimportant when editing, as it will snap to the centre of the hinge track at runtime.

Safe close-up

Once the player opens the cupboard, the safe is revealed and the tumblers on it's front can be manipulated. Each tumbler is a **Draggable** object, locked to a **Hinge Track**. The tumblers have no rotation limit: the Hinges all have **Is looped?** enabled. **Align drag vector to front?** Is also enabled, which makes manipulation easier when viewed from the front. The **Tumbler occluder** is a Collider that prevents the player from clicking on their far sides.

Snapping is enabled on each tumbler track, so that they snap to one of the numbered faces upon release. When the player lets go, the **LetGo: Tumbler** interaction is called, which determines whether or not the tumblers are in the correct place, and opens the door if they are. The tracks are child objects of the safe door, so that opening the door also moves the tumblers.

Tutorial tips

A series of tutorial tips appear to the top-left during gameplay, directing the player. These tips are displayed in the "Tutorial" Menu, which features a Label element that displays the value of the Global String Variable "Tutorial tip".

Different tips appear according to the player's progress. When a new tip is to be shown, a "parameterised" Cutscene, "Show tutorial tip" is called. This Cutscene turns off the menu, updates the variable according to a string parameter set by the ActionList: Run Action that called the Cutscene, and then turns the menu back on.

Sources

- Office chair model (CC0) by **hernandoramos**
- Dirty coffee machine model (CCA3) by **kednar**
- Lcd_monitor model (CCA3) by **bheema**
- Antique key model (CCA3) by **ducttapeman0**
- Pear Lowpoly model (CCA3) by **vitos1k**
- Low-Poly office phone model (CCA3) by **DennisH2010**
- Simple plant for game model (CCA3) by **myname**
- Five Rocks model (CC0) by **10000bones**
- low poly office table model (CCA3) by **Nmn9**
- "Scheming Weasel" music track (CCA3) by **Kevin MacLeod**