GameDev.tv

Game Jam Starter Kit

First Person Adventure Project

About

Thank you for downloading the GameDev.tv Game Jam 'First Person Adventure Project' Starter Kit!

This kit provides you the basic components you'll need to create a simple first person adventure game with a lock & key mechanic – allowing you to link certain objects (such as doors) to their corresponding pickups hidden in the world (such as keycards).

The kit including:

- first person character controller
- lock-and-key puzzle mechanic
- prototype assets and demo level

This document will guide you through the setup process and explain everything you need to get started!

Contents

Requirements

- Unity 2021.2 or later
- Input System 1.3 or later

Asset Overview

Once imported, you will find everything organized into several folders:

- Animations – Contains some simple animations for the doors
- Asset Packs – Includes several models from the Polygon Prototype pack, generously provided by Synty Studios
- Input – Contains the input settings and actions for the Unity Input System
- Materials – Contains some basic materials for the player and key/lock system
- Prefab – Includes the; player, keycard, door, and several environment prefabs
- Scenes – Includes the; "Sandbox" demo level
- Scripts – Contains the first person controller and key/lock scripts

Lock-and-Key Mechanic Explained

This project contains a simple lock-and-key system that can be used as the basis for some interesting puzzle mechanics in your game.

In the 'Sandbox' level provided, there are several colored doors that can only be opened once the player has collected the corresponding colored keycards.

You can change the `RemoveUsedKey` boolean toggle on the `Door.cs` component to specify whether a collected key should be removed once the corresponding door has been opened.

To add additional key type:

1. add the name to the public `KeyType` enum inside `enums.cs`
2. Add a new door to the scene and change the `DoorType` dropdown on the `Door.cs` component
3. Add a new KeyPickup to the scene and change the `KeyType` dropdown on the `KeyPickup.cs` component

Code Overview

All of the code is organized to be as readable as possible for beginners and additional comments have been added where necessary.

However, here’s a brief overview of the included classes.

FirstPersonLook.cs

This script is attached to the “First Person Camera” child game object of the “Player” prefab and is responsible for handling the look direction of the player.

Input is handled using the new Unity Input system and uses ‘Send Message’ rather than the more complex event driven system.

FirstPersonMovement.cs

This script is attached to the “Player” prefab and is responsible for handling the movement of the player.

Input is handled using the new Unity Input system and uses ‘Send Message’ rather than the more complex event driven system.

GroundCheck.cs

This script is attached to the “Ground Check” child game object of the “Player” prefab.
It is responsible for checking whether the player is on a surface and should be allowed to jump.

Note that if the child object is not present then `Jump.cs` will create it at runtime.

Jump.cs

This script is attached to the “Player” prefab and is responsible for handling the player jump ability.

Input is handled using the new Unity Input system and uses ‘Send Message’ rather than the more complex event driven system.

Door.cs

This script is attached to the “Door” prefab and allows you to configure what type of key will open it.

If the correct key ha been collected by the player, the door will open and the corresponding key will be removed from their inventory.

KeyPickup.cs

This script is attached to the “Keycard” prefab and allows you to configure what type of door it will open.

Once collected, the key will be added to the player inventory and the game object will be destroyed.

This script also allows you to configure whether the object moves/rotates in the game world, without needing the animator.

Inventory.cs

This script is attached to the “Player” prefab and stores which keys have been collected.

enum.cs

This script contains the public enum used by `Inventory.cs`, `Door.cs`, and `KeyPickup.cs`.

Need Help?

If you're not already part of our amazing community, why not come and say hi!

- Community Forum
- Discord
- Facebook
- Twitter
- Instagram
- YouTube

We also have a ton of great courses to help you develop your skills in; Unity, Blender, Pixel Art, and more.

Swing by GameDev.tv and supercharge your game development skills today!