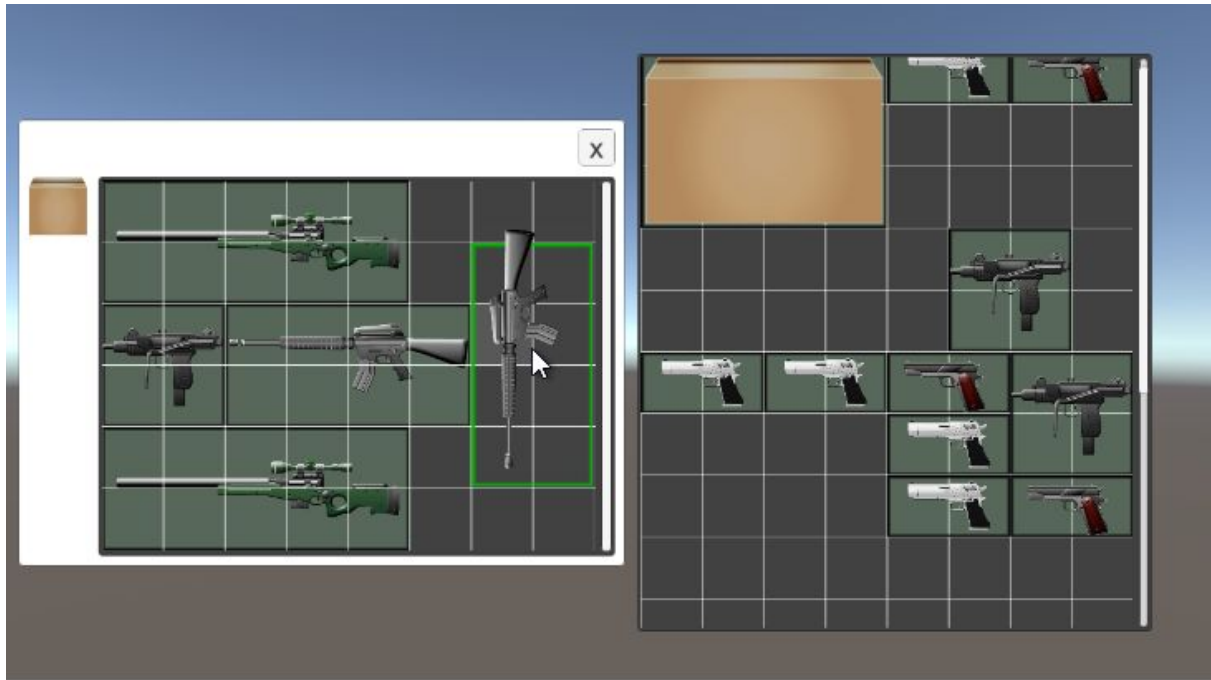


# Variable Inventory System



## はじめに

このドキュメントはUnity向けScript Asset **Variable Inventory System** のマニュアルです。  
このScript Assetは[無料](#)でGitHubもしくはAssetStoreで公開されています。

GitHub

<https://github.com/lisearcheleeds/VariableInventorySystem>

Unity AssetStore

<https://assetstore.unity.com/packages/slug/174066>

Copyright (c) 2020 Lise

Released under the MIT license

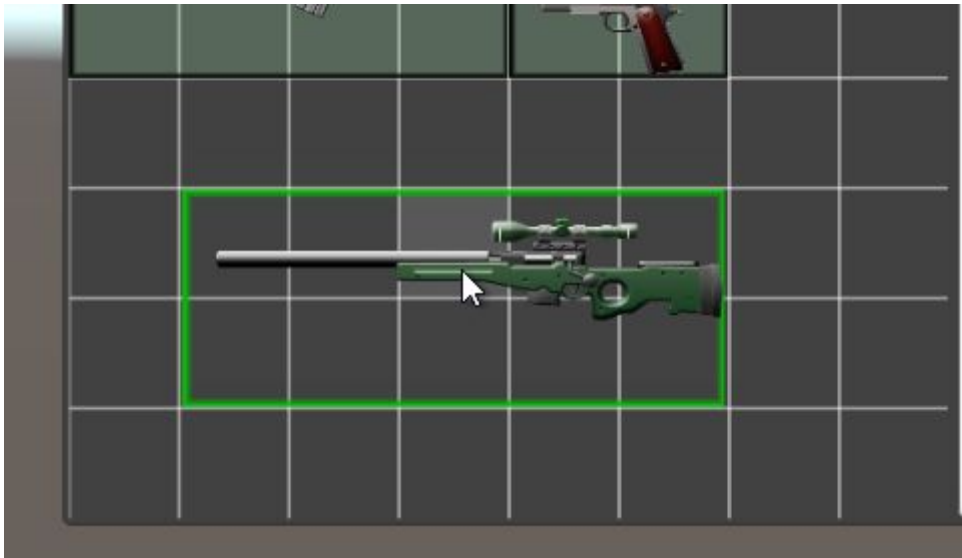
<https://github.com/lisearcheleeds/VariableInventorySystem/blob/master/LICENCE>

# Variable Inventory Systemとは

**Variable Inventory System** はアイテムの大きさを考慮したインベントリシステムです。このインベントリシステムは一般的なインベントリシステムのように1アイテム1セルではなく、1アイテムが複数のセルを利用します。

配置するアイテムはそれぞれがWidthとHeightそして回転しているかといった情報を持ちます。

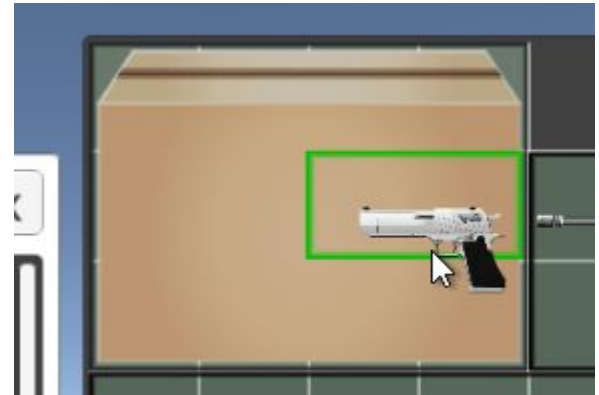
また、アイテムはプレイヤーのドラッグアンドドロップやインベントリ内に配置されているインベントリアイテムの上にドラッグアンドドロップすることでインベントリ領域間を移動させることができます。



## 特徴

1. アイテムはWidthとHeightを持ちます
  - a. インベントリ領域を飛び出る配置や、他のアイテムと重ねて配置することは出来ません
2. アイテムはCaseと呼ばれる種類のアイテムの上にドラッグアンドドロップすることでアイテムをそのCaseアイテム内に配置出来ます
  - a. Caseアイテム内のインベントリ領域を先端から探索していき、配置可能な位置に配置されます
3. アイテムをドラッグ中にRotateを呼び出すとそのアイテムが90度回転し、WidthとHeightが入れ替わります
4. インベントリ領域を継承したウィンドウを作ることで、一般的なゲームのプレイヤーの装備画面などが簡単に作ることが出来ます
5. スマートフォンの操作に対応しています

6. あなたのプロジェクトが独自に実装したボタン、リソース、リソースローダーを利用できます。



## クイックチュートリアル

UnityPackageには

1. Coreフォルダ
2. Standardフォルダ
3. Sampleフォルダ

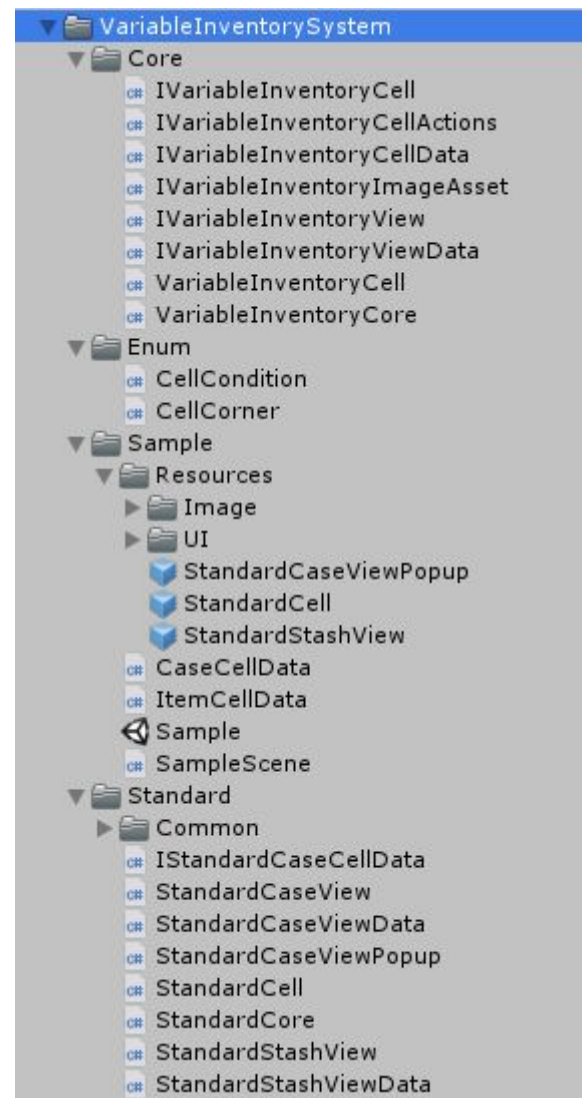
の順にプロジェクト依存度が高いものが入っています。

## Standardをそのまま利用する

StandardのScriptをそのまま利用する場合は  
**IVariableInventoryCellData** を継承したクラスを  
1つ実装する必要があります。

(Case機能を利用する場合は2つ実装します)

これらはSampleフォルダ内のSampleシーンとPrefabで確認できます。



# Simple Script Reference

- Core/VariableInventoryCore.cs
  - インベントリ領域である複数の IVariableInventoryView を管理します
  - ドラッグ中のアイテムを管理します
  - IVariableInventoryView から受け取ったイベントを複数の IVariableInventoryView に振り分けます
  - Standard実装
    - StandardCore
- Core/IVariableInventoryView.cs
  - 1つのインベントリ領域の表示を管理するインターフェイスです
  - 1つの IVariableInventoryViewData の内容を表示します
  - VariableInventoryCore から受け取ったイベントを元に適した振る舞いを行います
  - Standard実装
    - StandardStashView
    - StandardCaseView
- Core/IVariableInventoryViewData.cs
  - 1つのインベントリ領域のデータを管理するインターフェイスです
  - IVariableInventoryView から要求された配置可能位置や配置チェックを行います
  - Standard実装
    - StandardStashViewData
    - StandardCaseViewData
- Core/IVariableInventoryCell.cs
  - 1つのアイテムの表示を管理するインターフェイスです
  - 1つの IVariableInventoryCellData の内容を表示します
  - Cellは IVariableInventoryView を通して VariableInventoryCore にCellが起動した各イベントを返します
    - IVariableInventoryCellActions
  - Core実装
    - VariableInventoryCell
- Core/IVariableInventoryCellActions
  - IVariableInventoryCell の発火イベントを実装するためのインターフェイスです
    - onPointerClick
    - onPointerOptionClick
    - onPointerEnter
    - onPointerExit
    - onPointerDown
    - onPointerUp
- Standard/StandardStashView.cs
  - IVariableInventoryView の実装クラスです
  - シンプルなインベントリ表示を行います

- 1つの StandardStashViewData の内容を表示します
- Standard/StandardCaseView.cs
  - StandardCaseView を継承した IVariableInventoryView の実装クラスです
  - StandardCaseView からCase特有の例外処理を行います
  - 1つの StandardCaseViewData の内容を表示します
- Standard/StandardStashViewData.cs
  - IVariableInventoryViewData の実装クラスです
  - インベントリ領域内のアイテムの専有領域をmaskとして持ち、VariableInventoryCore からの要求に応じて処理を行います
- Standard/StandardCaseViewData.cs
  - StandardStashViewData を継承した IVariableInventoryViewData の実装クラスです
  - StandardStashViewData からCase特有の例外処理を行います