CarbonInput Library for Unity CheatSheet

Henrik Heine
InfectedBytes

October 1, 2017

1 Getting started

1. Download and import CarbonInput
2. Initialize Unity axis by clicking on:

   ```
   Edit > Project Settings > Carbon Input > Create Carbon Input Axis
   ```

3. Have fun with CarbonInput

If you want to use Unity's EventSystem you have to replace the `Standalone Input Module` with the `Carbon Input Module`.

2 Most important enumerations

Most methods have an optional `PlayerIndex` as a parameter. If you omit that parameter, `PlayerIndex.Any` will be used.

```
public enum PlayerIndex {
    Any, One, Two, Three, Four, Five, Six, Seven, Eight
}
```

```
public enum CButton {
    A, B, X, Y,
    Back, Start,
    LB, RB, // Left and Right bumper
    LS, RS, // Left and Right stick
}
```

```
public enum CAxis {
    LX, LY, // Left thumbstick
    RX, RY, // Right thumbstick
    LT, RT, // Left and Right trigger
    DX, DY  // DPad
}
```

```
public enum CStick {
    Left, Right, DPad
}
```

3 Basics

You can always skip the `PlayerIndex` parameter. If you do so, `PlayerIndex.Any` will be used.

```
public static class GamePad {
  public static bool GetButton(CButton btn, PlayerIndex id = PlayerIndex.Any) {...}
  public static float GetAxis(CAxis axis, PlayerIndex id = PlayerIndex.Any) {...}
  public static Vector2 GetStick(CStick stick, PlayerIndex id = PlayerIndex.Any) {...}
  public static Vector2 GetLeftStick(PlayerIndex id = PlayerIndex.Any) {...}
  public static Vector2 GetRightStick(PlayerIndex id = PlayerIndex.Any) {...}
  public static float GetLeftTrigger(PlayerIndex id = PlayerIndex.Any) {...}
  public static float GetRightTrigger(PlayerIndex id = PlayerIndex.Any) {...}
  public static Vector2 GetDPad(PlayerIndex id = PlayerIndex.Any) {...}
  public static GamePadState GetState(PlayerIndex id = PlayerIndex.Any) {...}
}
```

3.1 Buttons

```
// Check first gamepad
if(GamePad.GetButton(CButton.A, PlayerIndex.One)) { ... }
// Check any gamepad
if(GamePad.GetButton(CButton.A, PlayerIndex.Any)) { ... }
// same as:
if(GamePad.GetButton(CButton.A)) { ... }
```

For the sake of simplicity, we will skip the `PlayerIndex` in the next examples.

3.2 Single axis

```
// Get X axis:
float value = GamePad.GetAxis(CAxis.LX);
// Get left trigger
float value = GamePad.GetAxis(CAxis.LT);
// same as:
float value = GamePad.GetLeftTrigger();
```

3.3 Thumbsticks/DPad

You often want to have the x and y axis in one `Vector`:

```
// Get left thumbstick:
Vector2 left = GamePad.GetStick(CStick.Left);
// same as:
Vector2 left = GamePad.GetLeft();
// Get dpad:
Vector2 dpad = GamePad.GetStick(CStick.DPad);
// same as:
Vector2 dpad = GamePad.GetDPad();
```

3.4 States

You can also store the complete state of a gamepad:

```
GamePadState state = GamePad.GetState();
```

Now you can access all components more directly:

```
if(state.A) { ... }
// state.Left will give you a Vector2 representing the left thumbstick
controller.Move(state.Left);
```

In addition to the more direct way of accessing components, you can also determine now if a button was pressed or released during this frame:

```
// only true during the frame A was pressed
if(state.Pressed(CButton.A)) { ... }
// only true during the frame A was released
if(state.Released(CButton.A)) { ... }
```

4 Touch Controls

Not every mobile user owns a bluetooth gamepad, so you might want to add some nice onscreen buttons and joysticks.

1. Make sure you have a `Canvas` in your scene
2. Open the `Assets/CarbonInput/Prefabs` folder
3. Add any of those prefabs to your Canvas

5 Advanced

CarbonInput comes with a small settings file:
`Assets/CarbonInput/Resources/CarbonInput`.
When you click on the file you can change some settings inside the Inspector.
One of the most interesting settings is the `Behaviour` setting.

5.1 Behaviour

The `AnyBehaviour` enumeration defines the following three behaviours:

- `UseMappingOne`:
  Whenever `PlayerIndex.Any` is used, the system will get the motion of all connected gamepads, using the mapping of `PlayerIndex.One`.
  If all connected gamepads are using the same layout, this is the best solution. But if you have two controller with different layouts, like one XBox360 and one PS3 controller, this might cause problems, because they use different Unity Axes for their controls.
- `UseControllerOne`:
  The system will always use `PlayerIndex.One` instead of `PlayerIndex.Any`.
- `CheckAll`:
  Whenever `PlayerIndex.Any` is used, it will get the motion of all four gamepads and it will return the first match.

By default the behaviour is set to `CheckAll`.

5.2 Inverted Axis

By default the X axis of any axis will go from -1 (left) and +1 (right). Every Y axis will go from -1 (up) to +1 (down). So if you don't like that, you can just check the checkbox for the specific axis and the system will then flip the axis for you.