Dynamic D-pad

Manual

Manual version 1.0.0

Dynamic D-pad is a simple, responsive d-pad for games that use touch controls. Add the DynamicDpad component to a game object then read the axes values. It is as easy as that. There are properties exposed for tweaking, but the default values should work fine for most games. It automatically handles multiple screen resolutions and supports more than one d-pad at a time (e.g. dual stick controls).

Please note: The "DynamicDpad" class does not render the d-pad. You will have to use whatever GUI system you are using to render the d-pad. There is an example script "ExampleDpadRendering" that shows how to get the d-pad's coordinates for rendering.

How to run the demo scenes:

Add all the demo scenes to the build settings. Make sure the scene "Demo Menu" is the first in the list, if you want to build to a device. Open and play the scene "Demo Menu" if you want to test it in the editor. You will then be able to select scenes from the menu. You can return to the menu by clicking the "Menu" button at the top of the screen.

How to use the d-pad:

1) Add the component DynamicDpad to the player (or to any other game object).

2) In the player script, get a reference to the DynamicDpad component.

Example:

```
DynamicDpad dpad = gameObject.GetComponent<DynamicDpad>();
```

3) In the player script, read the axes values during the Update method.

Example 1:

```
float moveSpeed = 10.0f;
Vector3 velocity = new Vector3(dpad.Axes.x * moveSpeed * Time.deltaTime,
                       0.0f, dpad.Axes.y * moveSpeed * Time.deltaTime);
transform.position += velocity;
```

Example 2:
This example works best when “Use Axes Bias” is true on the DynamicDpad component.

```
float moveSpeed = 10.0f;
Vector3 pos = transform.position;
if (dpad.Axes.x < 0.0f)
{
      // Move the player left
      pos.x -= (moveSpeed * Time.deltaTime);
}
else if (dpad.Axes.x > 0.0f)
{
      // Move the player right
      pos.x += (moveSpeed * Time.deltaTime);
}
if (dpad.Axes.y < 0.0f)
{
      // Move the player down
      pos.z -= (moveSpeed * Time.deltaTime);
}
else if (dpad.Axes.y > 0.0f)
{
      // Move the player up
      pos.z += (moveSpeed * Time.deltaTime);
}
transform.position = pos;
```

A few notes on the camera:

If you need to get the d-pad's world coordinates for rendering, then you can set the “Gui Camera” property on the DynamicDpad component. A camera is needed to calculate the d-pad's world position. If you do not supply a camera then the component searches for the first active orthographic camera. If no orthographic camera is found then it uses the main camera (if a main camera has been tagged with the “MainCamera” tag).

Ideally, you should use an orthographic camera, instead of a 3D camera for rendering the d-pad. The 3D camera may move around and the d-pad's position will be drawn one frame behind. This will cause the d-pad to jump around on screen as the 3D camera moves. One way to avoid this is to add a component to the 3D camera and position the d-pad in the component's LateUpdate method, but there is still no guarantee that other game objects will not change the camera's position.