Simple Multiplatfom Gesture Controller.

Contents

Overview

Gesture Controller was created to simplify work with gestures. It's composed of C# scripts and extends standard Unity3D functional.
Currently support 6 platforms: iOS, Android, Windows Mobile, Windows, Mac OS, Linux. Plugin work equally at all those platforms.
This manual contents specifications of main plugin classes and few examples of code. Every examples from this document you can try in your Unity3d: you just need to import plugin and find appropriate scenes in Demo folder.

Specifications

Plugin don't need to be added for any GameObject, for initialize it you just need to call static gesture events from GestureController class.
All work with gestures consists of work with two classes: GestureController and Gesture which described below.

```
1.  /// Class, which implements tools for control gestures from touches and mouse.
2.  public class GestureController : Singleton<GestureController>
3.  {
4.          /// Gets the count of gestures.
5.          public static int CountOfGestures;
6.
7.          /// Gets or sets the gesture start delegate.
8.          /// Return null if GestureController haven't instance and it cannot be
    created
9.          public static func OnGestureStart;
10.
11.         /// Gets or sets the gesture end delegate.
12.         /// Return null if GestureController haven't instance and it cannot be
    created
13.         public static func OnGestureEnd;
14.
15.         /// Gets Gesture by it ID
16.         public static Gesture GetGByID(int id);
17.
```

```
        /// Stops the gesture.
        public static void StopGesture(Gesture g);
}

/// Information of gesture
public class Gesture
{
        /// This delegate call when gesture added information of new touch
        public func OnGestureStay;

        /// Get or set touches of gesture
        public List<GestureTouch> Frames;

        /// Return count of gesture touches
        public int FramesCount;

        /// Return ID of gesture or -1 if gesture haven't touches data
        public int ID;

        /// Return start point (in screen 2D space) of gesture
        public Vector2 StartPoint;

        /// Return end point (in screen 2D space) of gesture
        public Vector2 EndPoint;

        /// Calculate and return center point of gesture
        public Vector2 CenterPoint;

        /// Return angle of vector.
        private float Angle(Vector2 point);

        /// Return angle of first and last point of gesture
        public float FirsLastAngle();

```

```
        /// Calculate and return turns at gesture. Uses for process broken lines.
        public int TurnsCount(float minAngle,int countOfFrames);

        /// Get gesture code by translating every frame to numeric code
        public string Code;

        /// Get time of gesture
        public float GestureTime;

        /// Get length of gesture in screen pixels
        public float Distance;

        /// Gets the first touch
        public GestureTouch FirstTouch;

        /// Gets the last touch
        public GestureTouch LastTouch;
}
```

Examples

1. Example: Getting data of current gesture.

At this example i show how you can get all data of current gesture and write it to TextMesh in every point of time. I specifically did not added checking for different gestures, that will be showed in second example.

```
using UnityEngine;
using System.Collections;

[RequireComponent(typeof(TextMesh))]
public class CurrentGestureData : MonoBehaviour
{
```

```
7.          TextMesh _text;
8.
9.          /// <summary>
10.         /// Init
11.         /// </summary>
12.         void Awake()
13.         {
14.                 GestureController.OnGestureStart+=OnGestureStart;
15.                 GestureController.OnGestureEnd+=OnGestureEnd;
16.                 _text = GetComponent<TextMesh>();
17.         }
18.
19.         /// <summary>
20.         /// Finalize
21.         /// </summary>
22.         void OnDestroy()
23.         {
24.                 GestureController.OnGestureStart+=OnGestureStart;
25.                 GestureController.OnGestureEnd-=OnGestureEnd;
26.         }
27.
28.         /// <summary>
29.         /// Gets the gesture info.
30.         /// </summary>
31.         string GetGestureInfo(Gesture g)
32.         {
33.                 return string.Format("Current (last) gesture data:\r\nID [{0}]" +
34.            "\r\nStart point [{1}]\r\nCenter point [{2}]\r\nEnd point [{3}]\r\n" +
35.           "Gesture time [{4}]\r\nDistance [{5}]\r\nCode [{6}]" +
36.           "\r\nAngle between first/last points [{7}]\r\nFrames count [{8}]",
37.                                                         g.ID,
38.                                                         g.StartPoint,
39.                                                         g.CenterPoint,
40.                                                         g.EndPoint,
```

```
                                                        g.GestureTime,
                                                        g.Distance,
                                                        g.Code,
                                        g.FirsLastAngle(),
                                                        g.FramesCount);
        }

        /// <summary>
        /// Raises the gesture start event.
        /// </summary>
        void OnGestureStart (Gesture g)
        {
                g.OnGestureStay += OnGestureStay;
                _text.text = GetGestureInfo(g);
        }

        /// <summary>
        /// Raises the gesture stay event.
        /// </summary>
        void OnGestureStay (Gesture g)
        {
                _text.text = GetGestureInfo(g);
        }

        /// <summary>
        /// Raises the gesture end event.
        /// </summary>
        void OnGestureEnd (Gesture g)
        {
                _text.text = "";
        }
}
```

2. Example: Throw an object with a gesture

In this example i will show how you can throw any 2d object in scene, which has layer from ThrowMask. This example contains part, which detecting ID of gesture and you can throw only one object in one time. And this example will show how you can stop every gesture in any time: this will be useful when you need to control length of gesture or time of gesture.

```
1.  using UnityEngine;
2.  using System.Collections;
3.
4.  // Need to add at object with camera
5.  [RequireComponent(typeof(Camera))]
6.  public class Throw2dObject : MonoBehaviour
7.  {
8.          public float ThrowForce = 400;
9.          // layer mask of objects which can be throwed
10.         public LayerMask ThrowMask;
11.
12.         // current throw target
13.         private Transform _target;
14.         // current gesture ID
15.         private int _gID = -1;
16.
17.         // Init
18.         void Awake()
19.         {
20.                 GestureController.OnGestureStart+=OnGestureStart;
21.                 GestureController.OnGestureEnd+=OnGestureEnd;
22.         }
23.
24.         // Finalize
25.         void OnDestroy()
26.         {
27.                 // It's need to safely load enother level
28.                 GestureController.OnGestureStart+=OnGestureStart;
```

```
29.                 GestureController.OnGestureEnd-=OnGestureEnd;
30.         }
31.
32.         /// Raises the gesture start event.
33.         void OnGestureStart (Gesture g)
34.         {
35.                 // will remember gesture ID
36.                 if (_gID == -1)
37.                 {
38.                         // if gesture start from any object
39.                         Ray r = camera.ScreenPointToRay(g.StartPoint);
40.                         RaycastHit2D hit =
    Physics2D.Raycast(r.origin,r.direction,100,ThrowMask);
41.                         if (hit)
42.                         {
43.                                 // remember it
44.                                 _target = hit.rigidbody.transform;
45.                                 g.OnGestureStay += OnGestureStay;
46.                                 _gID = g.ID;
47.                         }
48.                 }
49.         }
50.
51.         /// Raises the gesture stay event.
52.         void OnGestureStay (Gesture g)
53.         {
54.                 // for example, we need only short gestures
55.                 // (about 0.1 seconds)
56.                 if (g.GestureTime > 0.1f && g.StartPoint != g.EndPoint)
57.                         GestureController.StopGesture(g);
58.         }
59.
60.         /// Raises the gesture end event.
61.         void OnGestureEnd (Gesture g)
```

```
62.         {
63.                 // if it our membered gesture
64.                 if (_gID == g.ID)
65.                 {
66.                         if (_target != null && _target.rigidbody2D != null)
67.                         {
68.                                 Vector3 start =
    camera.ScreenToWorldPoint(g.StartPoint);
69.                                 Vector3 end =
    camera.ScreenToWorldPoint(g.EndPoint);
70.                                 // than throw it
71.                                 _target.rigidbody2D.AddForce(ThrowForce * (end -
    start).normalized);
72.                         }
73.                         _gID = -1;
74.                 }
75.         }
76. }
```

Contacts

I'll be happy to get any feedback about this plugin or comments about its development away.
With best wishes, Slaboshpitsky Radomir, creator of GestureController.
E-mail: radiys92@gmail.com