Gesture Recognition

Hello, welcome to readme, which contains basic instructions how to use gesture recognition asset.

About

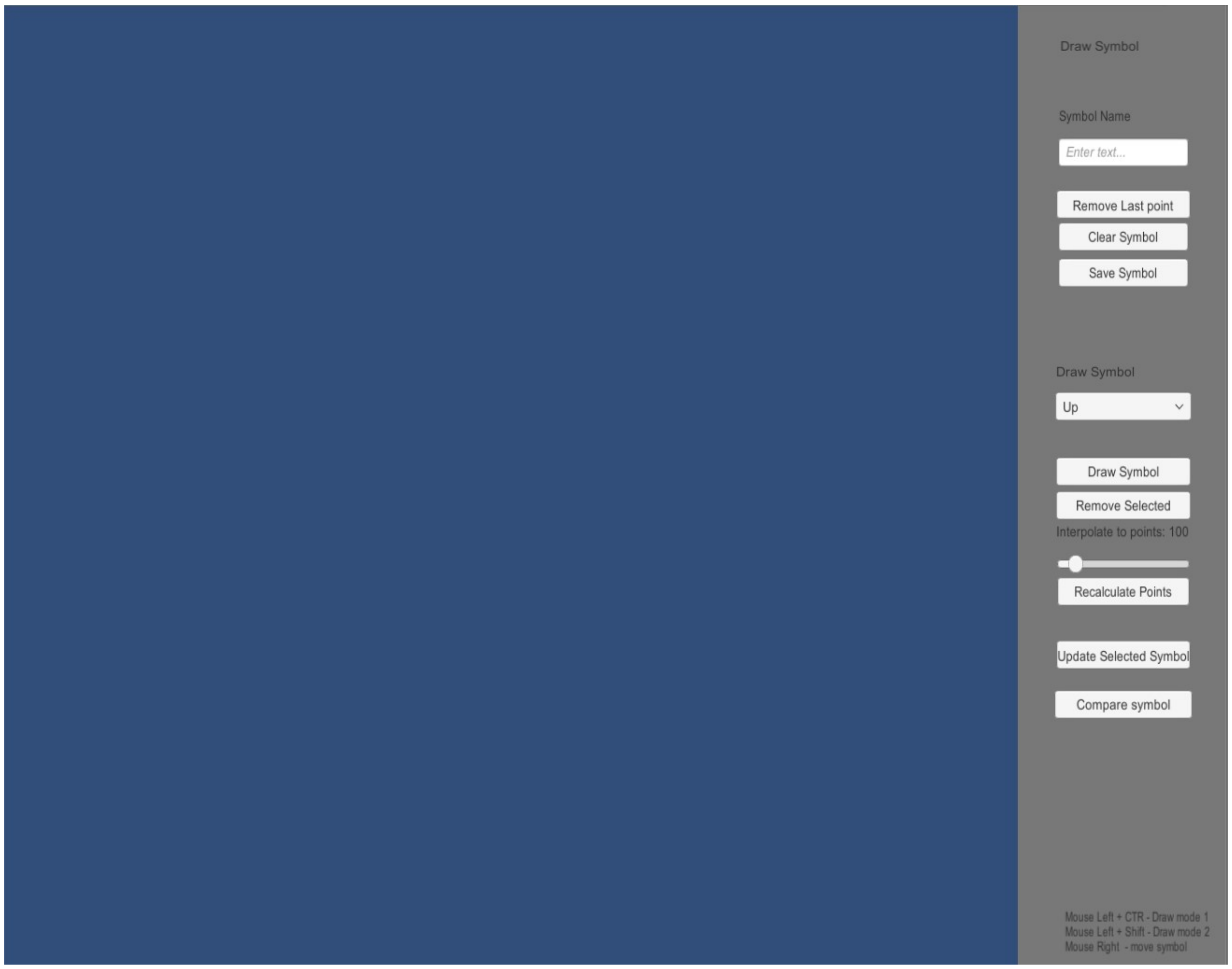
This asset was created for my own project where I needed good capture recognition tool which can handle different size of gestures and gesture need to be sensitive about direction.

Current state of asset can be used for fast and relative precise gesture recognition on computers (mobile input can be extended quite easily)

Manager Mode

Asset came with set of existing gestures but user can easily added new ones or removes old ones

Editor scene (under scene name: **SymbolManagmentScene**) looks like this



Basic controls

Mouse left click + CTR - Drawing gestures

Mouse left click + SHIFT – Drawing gestures with delay

Mouse right click – move gesture

Right panel contains all option for user

1) Textfield for new gesture name
2) Remove Last Point: remove last draw point of current gesture (for correcting mistakes)
3) Clear symbol: Clear current symbol from screen
4) Save symbol: save current symbol with name in textfield (point 1)
5) Drop down list with existing symbols
6) Draw symbol: draw symbol selected in drop down list
7) Recalculate points: For existing (saved) symbols from drop down list enable “interpolate” points to check how gesture looks like with different number of points. This is for correct setting of precision parameter for your game. (Better precision means lower performance so it needs to be set best for your types of gestures and target machines)
8) Update Selected Symbol: when you are editing existing symbols (move, add points, remove points)
9) Compare symbol: test function to give you comparison. Currently gives you 2 types of comparison. Default one and advance one which trying to remove as many gestures as possible as insufficient ones (should be faster and in most cases better)
10) Label with few instruction

Symbol Comparison

Comparison is done by comparing angles between points in symbol. Therefore comparison not depends on size (small or big rectangle is same). But depend on start and end point and direction of drawing.

For example horizontal line from left to right is not same as from light to left. Also if start draw square from top right corner and next time from top left corner (both clockwise) algorithm will not recognize these two squares as similar.

For better seeing direction points are colored.

Demo Scene

Contains one script (**MhGestureGameCapture)** which is enough for capturing gestures in your game.

MhGestureGameCapture .GestureIsDrawing can be edited for different drawing command than left mouse button.

Use basic version of gesture recognition (advance ones can be found in **MhGestureInput.CompareGesture**)

Tips

Precision range is from 0 -100%, but in reality only gestures above 60-70% up are really similar enough. Same from the other side it is hard to get 99%+ precision so most time 95%+ is almost perfect.

Gestures are saving into **Gesture.json** file, default location is **Assets/MhGesture/Data/Gestures.json** . You need to copy this file with your game every time.

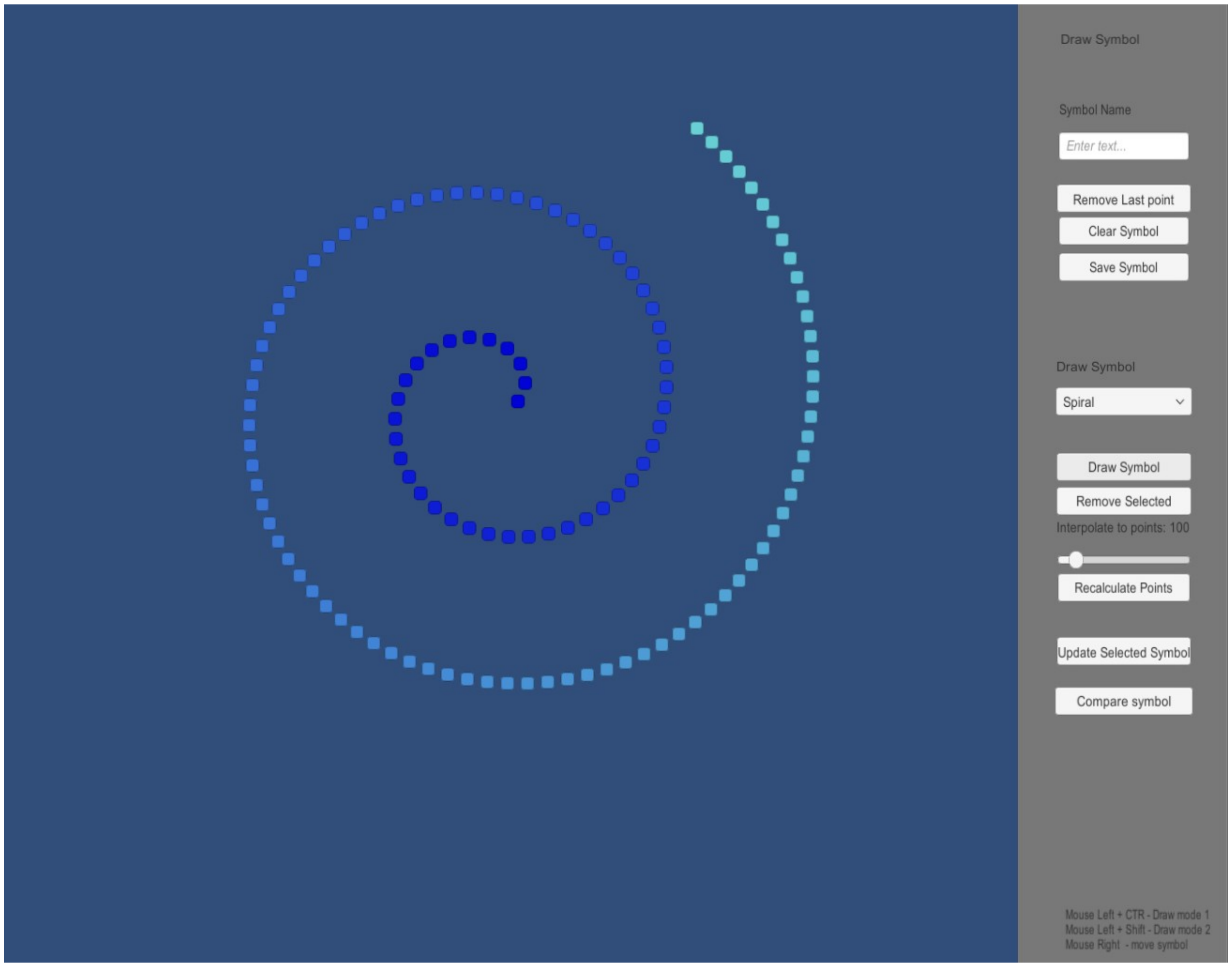
If you cant draw perfect gestures (like circle) by hand, you can create points in code and add them manually (with name) into json file, it is better for most cases.

Easy customization properties

MhGestureManager.distanceConstant – count basic distance between points when user drawing (higher more precise)
MhGestureManager.gestureCaptureEnable – Boolean can globally enable or disable capturing gesture in game.

MhGestureManager.minNumberOfpoints - minimum number of points to recognize gesture, and not compare every click, (in most cases can be higher)

MhGestureManager.analyzer – this analyzer has default number of points 500 which means every symbol will be interpolated into 500points length symbol and then compare. Higher number of points could leave to stricter comparison but little bit more performance heavy. Too few points could also leads to incorrect comparison.

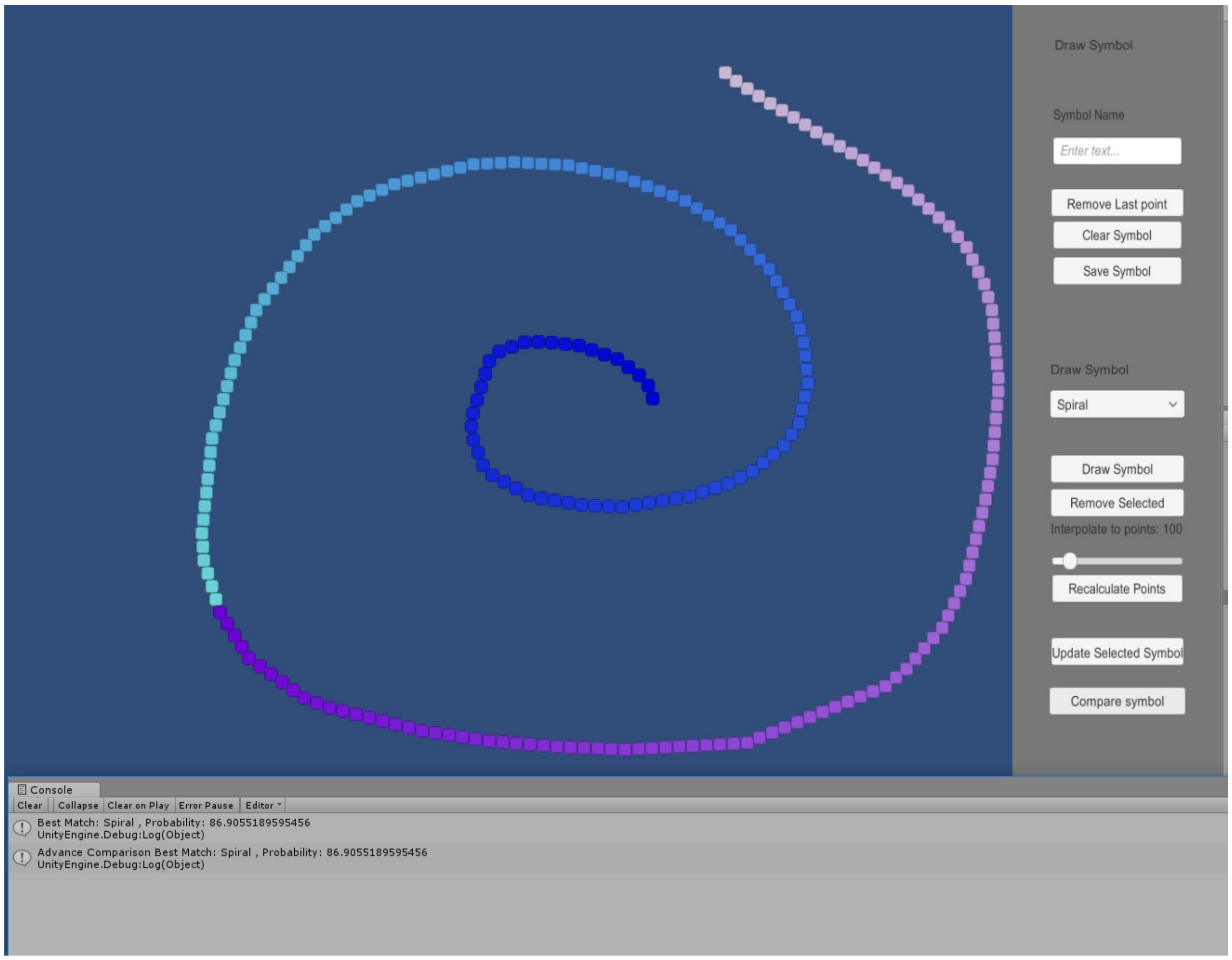


Symbol Name: Enter text...

Draw Symbol: Spiral

Interpolate to points: 100

Mouse Left + CTR - Draw mode 1
Mouse Left + Shift - Draw mode 2
Mouse Right - move symbol



```
Best Match: Spiral , Probability: 86.9055189595456
UnityEngine.Debug:Log(Object)
Advance Comparison Best Match: Spiral , Probability: 86.9055189595456
UnityEngine.Debug:Log(Object)
```