

Kinect Gesture Controller

Οδηγίες εγκατάστασης

Αρχικά χρειάζεται το Kinect For Windows 2.0 SDK που βρίσκεται στον παρακάτω [σύνδεσμο](https://go.microsoft.com/fwlink/p/?LinkId=513177) (<https://go.microsoft.com/fwlink/p/?LinkId=513177>).

Από τον Unity 3D Editor επιλέγουμε *Assets => Import Package => Custom Package*. Έπειτα το αρχείο Kinect.2.0.1410.19000.unitypackage του SDK κι έπειτα το κουμπί *Import*. Τα ίδια βήματα ισχύουν και για το .unitypackage της βιβλιοθήκης.

Αφού ολοκληρωθεί το import του SDK και της βιβλιοθήκης, πάμε στη σκηνή που θα χρησιμοποιηθεί η βιβλιοθήκη και επιλέγουμε από τη γραμμή του μενού *Kinect => Create Kinect Gesture Manager*.

ΠΡΟΣΟΧΗ!

Ο Kinect Gesture Manager είναι υποχρεωτικό να υπάρχει σε κάθε σκηνή που θέλουμε να χρησιμοποιηθεί, αλλιώς δεν θα υπάρχει ανίχνευση της εισόδου από τη συσκευή Microsoft Kinect 2.0.

Όλες οι συναρτήσεις το *GestureManager* περιέχουν το όρισμα *int personId* το οποίο αναφέρεται στον χρήστη που θέλουμε να αναφερθούμε. Αν δεν δώσουμε κάποιο έχει προκαθορισμένη τιμή 1. Αν η εφαρμογή αναφέρεται μόνο σε ένα χρήστη μπορούμε να το αγνοήσουμε.

Τα csv αρχεία καταγραφής του χεριού δείκτη αποθηκεύονται στην τοποθεσία **StreamingAssets/CSV/** της εφαρμογής μετά το τέλος κάθε καταγραφής με το όνομα αρχείου στη μορφή *PlayerX YYYY-MM-DD HH-MM-SS*.

Οδηγίες χρήσης

Παρακάτω υπάρχουν κομμάτια κώδικα για το πως χρησιμοποιείται η βιβλιοθήκη.

```
if (GestureManager.GetGestureStatus(GestureId.SwipeUpRightHand))
{
    Debug.Log("Swipe up right hand!");
}
```

Όταν η συνάρτηση *GestureManager.GetGestureStatus* επιστρέφει true το gesture έχει εντοπιστεί στο εκάστοτε frame. Ως όρισμα δέχεται ένα από τα προκαθορισμένα *GestureId* της βιβλιοθήκης.

```
Vector3 handposWorld = GestureManager.GetHandPointerWorldPos();  
_cube.transform.position = handposWorld;
```

Η συνάρτηση `GestureManager.GetHandPointerWorldPos` επιστρέφει ένα `Vector3` με την τιμή της θέσης του χεριού που βρίσκεται πιο κοντά στη συσκευή Microsoft Kinect.

```
if (!_recording && GestureManager.GetGestureStatus(  
    GestureId.WaveRightHand))  
{  
    Debug.Log("Start recording");  
    GestureManager.StartRecordingHandPositions();  
    _recording = true;  
}  
  
if (_recording && GestureManager.GetGestureStatus(GestureId.WaveLeftHand))  
{  
    Debug.Log("Stop recording");  
    GestureManager.StopRecordingHandPositions();  
    _recording = false;  
}
```

Όταν εντοπίζεται το gesture με `GestureId.WaveRightHand` και δεν έχει ξεκινήσει η καταγραφή τότε ξεκινά η καταγραφή των θέσεων του χεριού δείκτη.

Όταν εντοπίζεται `GestureId.WaveLeftHand` και έχει ήδη ξεκινήσει η καταγραφή τότε η καταγραφή των θέσεων του χεριού δείκτη σταματάει και αποθηκεύει σε ένα αρχείο csv στην τοποθεσία **StreamingAssets/CSV/** τα δεδομένα που καταγράφηκαν με όνομα αρχείου στη μορφή *PlayerX YYYY-MM-DD HH-MM-SS*.

```
if (GestureManager.CursorHandStateStart(HandState.Closed))  
{  
    Debug.Log("Hand Closed Start");  
}  
  
if (GestureManager.CursorHandStateEnd(HandState.Closed))  
{  
    Debug.Log("Hand Closed End");  
}
```

Όταν η συνάρτηση `GestureManager.CursorHandStateStart` επιστρέφει `true` τότε στο εκάστοτε frame τότε το χέρι που λειτουργεί εκείνη τη στιγμή ως δείκτης ξεκίνησε να βρίσκεται στην επιθυμητή κατάσταση `HandState` που έχουμε ορίσει.

Όταν η συνάρτηση `GestureManager.CursorHandStateEnd` επιστρέφει `true` τότε στο εκάστοτε frame τότε το χέρι που λειτουργεί εκείνη τη στιγμή ως δείκτης ξεκίνησε να παύει στην επιθυμητή κατάσταση `HandState` που έχουμε ορίσει.

Αυτές οι συναρτήσεις μπορούν να χρησιμοποιηθούν για τη μετακίνηση ενός αντικειμένου στο χώρο. Παράδειγμα αυτής της λειτουργίας υπάρχει στο Demo2D στη βιβλιοθήκη.

Static Public Member Functions

static TrackingConfidence	GetCursorHandConfidence (int personId=1) Gets the confidence of the current cursor hand. More...
static bool	HandStateStart (HandState handState, Person.Hand hand, int personId=1) Notify when the requested hand state started. More...
static bool	CursorHandStateStart (HandState handState, int personId=1) Notify when the requested cursor's hand state starts. More...
static bool	HandStateEnd (HandState handState, Person.Hand hand, int personId=1) Notify when the requested hand state ended. More...
static bool	CursorHandStateEnd (HandState handState, int personId=1) Notify when the requested cursor's hand state stopped. More...
static bool	GetGestureStatus (GestureId gestureId, int personId=1) Get the status of a gesture for a specific player. More...
static HandState	GetCursorHandState (int personId=1) Gets cursor hand state of current frame. More...
static HandState	GetCursorHandStatePrevious (int personId=1) Gets cursor hand state of previous frame. More...
static HandState	GetHandState (Person.Hand hand, int personId=1) Get the state of person's hand. More...
static HandState	GetHandStatePrevious (Person.Hand hand, int personId=1) Get the state of person's hand on previous frame. More...
static Vector2	GetHandPointer (int personId=1) Get the hand position in screen space. More...
static Vector3	GetHandPointerWorldPos (int personId=1) Get the hand position in the world space. More...
static void	StartRecordingHandPositions (int personId=1) Checks if person's index is correct and starts recording hand's position. More...
static void	StopRecordingHandPositions (int personId=1) Checks if person's index is correct and stops recording hand's position. More...

Public Attributes

int	MaximumPlayers = 1 Maximum people allowed to be recognized by the Kinect sensor. More...
float	PauseBetweenGestures Delay between gesture recognitions. More...
bool	DebugMode Print messages on console when a gesture recognized. More...
bool	BypassForbiddenGestures Ignores the forbidden gesture check. More...
float	RecordingDelay = 0.5f