LeapC API

*Ce document est destiné à fournir des explications sur le fonctionnement de l’API du Leap Motion*

La librairie LeapC agit comme un intermédiaire entre le micrologiciel du Leap Motion et l’application. Une file d’attente est ainsi mise en place dans laquelle sont placés les données de tracking, les images et des messages de statut.

Une fois le service du Leap Motion démarré, la fonction [LeapPollConnection()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga2a8aecad339f0fd339ca22a3e7b389f6) va retourner à chaque appel le prochain message dans la file d’attente.

# Utiliser l’API LeapC

## Démarrage du service Leap Motion

Pour récupérer les messages envoyés par le Leap Motion il faut initialiser une méthode qui tournera sur son propre thread en continu. On fera appel à la fonction [LeapPollConnection()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga2a8aecad339f0fd339ca22a3e7b389f6).

|  |
| --- |
| void serviceMessageLoop(){  [eLeapRS](https://developer.leapmotion.com/documentation/v4/group___enum.html#ga10647f52cdf6742a654aab0054ce3d3e) result;  [LEAP\_CONNECTION\_MESSAGE](https://developer.leapmotion.com/documentation/v4/group___structs.html#struct_l_e_a_p___c_o_n_n_e_c_t_i_o_n___m_e_s_s_a_g_e) msg;  while(\_isRunning){  unsigned int timeout = 1000;  result = [LeapPollConnection](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga2a8aecad339f0fd339ca22a3e7b389f6)(connectionHandle, timeout, &msg);  //Handle message  }  } |

*La liste des types que la fonction LeapPollConnection() peut retourner est décrite dans une énumération :* [*https://developer.leapmotion.com/documentation/v4/group\_\_\_enum.html#ga04f93b375f7c8040178ec5be2bf658ec*](https://developer.leapmotion.com/documentation/v4/group___enum.html#ga04f93b375f7c8040178ec5be2bf658ec)

Une fois cette fonction définie, nous pouvons lancer la “pompe à messages” :

|  |
| --- |
| LEAP\_CONNECTION\* OpenConnection(){  [eLeapRS](https://developer.leapmotion.com/documentation/v4/group___enum.html#ga10647f52cdf6742a654aab0054ce3d3e) result = [LeapCreateConnection](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga5bcc831cf503136f45dde040f29973b5)(NULL, &connectionHandle);  if(result == [eLeapRS\_Success](https://developer.leapmotion.com/documentation/v4/group___enum.html#gga10647f52cdf6742a654aab0054ce3d3ea0ea6fda288d9d060eac60e2d7e7adb0e)){  result = [LeapOpenConnection](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga49a90e9ca6d880d59fa3f1813689f7fd)(connectionHandle);  if(result == [eLeapRS\_Success](https://developer.leapmotion.com/documentation/v4/group___enum.html#gga10647f52cdf6742a654aab0054ce3d3ea0ea6fda288d9d060eac60e2d7e7adb0e)){  \_isRunning = true;  pthread\_create(&pollingThread, NULL, serviceMessageLoop, NULL);  }  }  return &connectionHandle;  } |

*LeapOpenConnection() va effectuer une vérification côté client ainsi que les opérations nécessaires au démarrage du service Leap Motion*

## Interpolation des images

A la place de récupérer les images du Leap Motion à chaque fois qu’elle sont générées, nous pouvons choisir de ne récupérer des images qu’à un interval de temps donné. LeapC va donc interpoler l’image la plus proche à ce temps donné pour générer l’image.

En général, demander des images plus loin dans le temps vont améliorer la sensation de fluidité mais augmenter la latence. Il faut donc trouver le compromis idéal pour l’application (ex: entre sur mobile et sur pc les valeurs vont changer pour la même application).

Si l’interval de temps choisi est plus petit que celui de la fonction [LeapPollConnection()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga2a8aecad339f0fd339ca22a3e7b389f6), LeapC va extrapoler les images à la place de les interpoler.

La fonction [LeapGetNow()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga4ef33708af974ecd618ad9784aa38161) va retourner l’horloge interne utilisé pour générer l’interval de temps. Cela peut permettre de recalculer l’horloge désirée.

Pour récupérer les images interpolées :

|  |
| --- |
| 1. Open the connection. 2. Call [LeapPollConnection()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga2a8aecad339f0fd339ca22a3e7b389f6) at least once to complete opening the connection. 3. Create a [LEAP\_CLOCK\_REBASER](https://developer.leapmotion.com/documentation/v4/group___structs.html#struct_l_e_a_p___c_l_o_c_k___r_e_b_a_s_e_r) struct. 4. Pass this struct to [LeapCreateClockRebaser()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga95c0d7e31b8337021c41f13d4ef6849b), which initializes the struct. 5. Every time you render a frame of graphics, call [LeapUpdateRebase()](https://developer.leapmotion.com/documentation/v4/group___functions.html#gac111f105a4b418e1f7ed08a1b74c8bca) to synchronize the Leap Motion and application clocks. The clocks will drift apart rapidly, otherwise. 6. Translate the target timestamp of the interpolated frame by passing the application time to [LeapRebaseClock()](https://developer.leapmotion.com/documentation/v4/group___functions.html#gadd9e1af6480d7948b77ccf40fbca337a). LeapC writes the corresponding Leap clock time to the pLeapClock parameter. 7. Call [LeapGetFrameSize()](https://developer.leapmotion.com/documentation/v4/group___functions.html#gae680ca44ccf77a25c4a61f9ae1a311bc) to calculate the size of the buffer needed to require the tracking data at the desired point in time. (Frame size varies with the number of hands in view.) 8. Allocate a buffer of the required size. 9. Call [LeapInterpolateFrame()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga7cbdc29069fbcd6aca1a16989722e85c), passing in the frame time and your buffer. LeapC writes the frame data to your buffer. This buffer can be treated as a [LEAP\_TRACKING\_EVENT](https://developer.leapmotion.com/documentation/v4/group___structs.html#struct_l_e_a_p___t_r_a_c_k_i_n_g___e_v_e_n_t) object to access the tracking data. 10. Continue calling [LeapUpdateRebase()](https://developer.leapmotion.com/documentation/v4/group___functions.html#gac111f105a4b418e1f7ed08a1b74c8bca) and [LeapInterpolateFrame()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga7cbdc29069fbcd6aca1a16989722e85c) at your application frame rate. |

*Exemple de code d’interpolation :*

[*https://developer.leapmotion.com/documentation/v4/interpolation-example.html\**](https://developer.leapmotion.com/documentation/v4/interpolation-example.html*)

## Récupérer les informations de l’appareil

Vous pourrez avoir besoin des informations matérielles du Leap Motion. La démarche est décrite ici : <https://developer.leapmotion.com/documentation/v4/usingleapc.html#devices>

## Modifier la *politique* du Leap Motion

Les *politiques* vont modifier le comportement ou l’utilisation du Leap Motion (Par exemple choisir entre un dispositif posé sur un bureau ou monté sur un casque).

Les politiques disponibles sont décrites ici : <https://developer.leapmotion.com/documentation/v4/concepts.html#policy_concept>

Pour modifier la politique utilisée :

|  |
| --- |
| 1. Open the connection. 2. Call [LeapPollConnection()](https://developer.leapmotion.com/documentation/v4/group___functions.html#ga2a8aecad339f0fd339ca22a3e7b389f6) at least once. 3. Combine the flags that you want to set by bitwise ANDing elements of the eLeapPolicyFlag enumeration. 4. Likewise, combine the flags that you want to clear by bitwise ANDing elements of the eLeapPolicyFlag enumeration. 5. Call [LeapSetPolicyFlags()](https://developer.leapmotion.com/documentation/v4/group___functions.html#gab57050814a0763ec07ed088e3d2de7f2), passing in your set and clear policy combinations. 6. When the policy change is processed by the service, LeapC adds a [LEAP\_POLICY\_EVENT](https://developer.leapmotion.com/documentation/v4/group___structs.html#struct_l_e_a_p___p_o_l_i_c_y___e_v_e_n_t) message to the queue. You can compare [LEAP\_POLICY\_EVENT::current\_policy](https://developer.leapmotion.com/documentation/v4/group___structs.html#a15e97d53b785691e29631b22a4b70119) with the desired policy to verify that the policy change was "approved." |

## Fichiers de configuration

Il est important de jeter un oeil aux fichiers de configuration. Ceux-ci agissent aussi sur le comportement du Leap Motion mais à un niveau encore plus bas que les politiques. Vous trouverez ci-dessous les liens vers les pages de documentation liées.

La liste des paramètres modifiables est présente sur cette page : <https://developer.leapmotion.com/documentation/v4/configuration.html>

La démarche pour lire et modifier un fichier de configuration est décrite ici :

<https://developer.leapmotion.com/documentation/v4/usingleapc.html#config>

# Images du LeapMotion

Le LeapMotion utilise deux caméras infrarouges pour récupérer les informations de tracking des mains. L’API LeapC fournit ainsi un buffer contenant les valeurs de luminosité des capteurs et un autre buffer contenant une map de calibration de la caméra utilisée pour corriger la distorsion des données de l’image.