



murmur

|ℳ| documentation murmur

Document de lancement et descriptif du software MURMUR

page 3. Raccourcis clavier :

page 4. Glossaire :

page 5. Installation :

page 6. Configuration de départ :

page 10. Prise en main de l'interface :

page 12. Écran de projection :

page 14. Animation :

page 18. Configuration :

page 21. Devices :

page 25. Network :

page 27. Sound :

page 30. Surface :

|▽| raccourcis clavier

F : Fullscreen

Esc : Quitter l'application

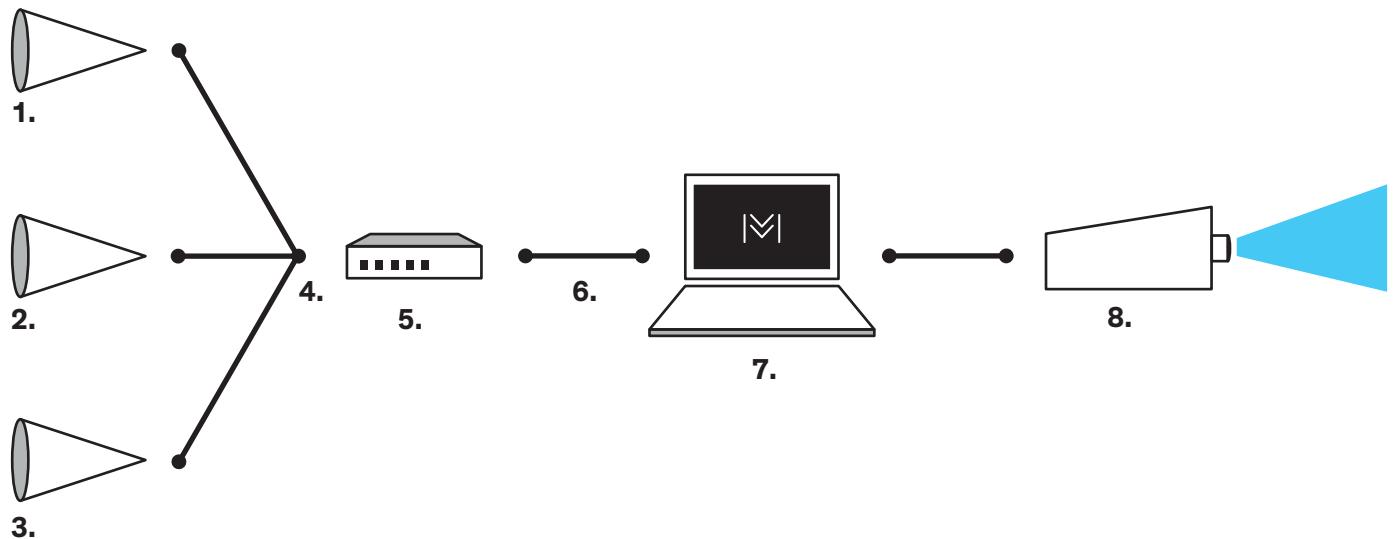
Espace : Écran de projection

Flèches directionnelles : gauche/droite Changer d'animations

|V| glossaire

Device :	Chambre d'écho ou cône de captation sonore.
Fichier xml :	Ou eXtensible Markup Language, est un fichier de configuration textuel.
Ledstrip :	Ruban de LED.
Mapping :	Technique qui consiste à déformer une projection afin de l'adapter à la surface désirée.
Madmapper :	Logiciel de mapping.
Raspberry pi :	Le Raspberry Pi est un nano-ordinateur de la taille d'une carte de crédit.
Soft :	Soft ou Software désigne un logiciel.
Fullscreen :	Plein écran.
Quadwarping :	Technique basique de mapping.
Cpp :	Fichier écrit dans le langage C++.
Js :	Fichier écrit dans le langage Javascript.
Playlist :	Liste de lecture.
Mode standby :	Mode veille
Network :	Réseaux
FBO :	Frame Buffer Objects, surface où sont dessinées les animations.
Drag & drop :	Glisser déposer, principe de déplacement de fichier par le biais de la souris.

|ℳ| installation



1: Chambre d'écho n°1 avec sa propre adresse IP: 10.23.108.139.

2: Chambre d'écho n°2 avec sa propre adresse IP: 10.23.108.140.

3: Chambre d'écho n°3 avec sa propre adresse IP: 10.23.108.141.

4: Communication avec le routeur sur le port: 1235.

5: Routeur.

6: Liaison Ethernet routeur/ordinateur.

7: Ordinateur avec sa propre adresse IP communiquant avec le routeur sur le port: 1234.

8: Vidéoprojecteur.

|M| configuration de départ

Le lancement de MURMUR nécessite d'ouvrir un fichier XML sur lequel seront indiquées certaines informations et certains réglages permettant le calibrage et l'utilisation de MURMUR. Ce fichier XML permet principalement de préciser si MURMUR est lancé en MURMUR simulation ou en mode installation dans le but d'une présentation devant un public.

```

<?xml version="1.0" encoding="UTF-8"?>
<murmur>
    <simulator>
        <!-- Activate or not simulation -->
        <enable>1</enable>

        <!-- Sound input for simulators -->
        <soundInput>
            <device><!-->
                <nbChannels>1</nbChannels>
            </device>
        </soundInput>

        <!-- Devices -->
        <devices>
            <device>
                <id>deviceEchoSimulator01</id>
                <nblEDs>158</nblEDs>
                <length>4.0</length>
            </device>

            <!-->
            <device>
                <id>deviceEchoSimulator02</id>
                <nblEDs>158</nblEDs>
                <length>2.0</length>
            </device>
        </devices>
    </simulator>

    <!-- MadMapper -->
    <madmapper>stereolux.mad</madmapper>

    <!-- Raspberry -->
    <launchDevices>
        <ip>10.23.108.139</ip>
    </launchDevices>

```

1. Lancement de MURMUR, fichier de configuration : /data/configuration.xml

Fichier xml permettant d'intervenir manuellement par le biais d'un éditeur de texte sur les réglages de bases de l'application MURMUR.

```

<?xml version="1.0" encoding="UTF-8"?>
<murmur>
    <simulator>
        <!-- Activate or not simulation -->
        <enable>1</enable>

        <!-- Sound input for simulators -->
        <soundInput>
            <device><!-->
                <nbChannels>1</nbChannels>
            </device>
        </soundInput>

        <!-- Devices -->
        <devices>
            <device>
                <id>deviceEchoSimulator01</id>
                <nblEDs>158</nblEDs>
                <length>4.0</length>
            </device>

            <!-->
            <device>
                <id>deviceEchoSimulator02</id>
                <nblEDs>158</nblEDs>
                <length>2.0</length>
            </device>
        </devices>
    </simulator>

    <!-- MadMapper -->
    <madmapper>stereolux.mad</madmapper>

    <!-- Raspberry -->
    <launchDevices>
        <ip>10.23.108.139</ip>
    </launchDevices>

```

2. Commande permettant d'indiquer si l'utilisation de MURMUR est une simulation ou une installation.

<enable>1</enable>: simulateur

<enable>0</enable>: installation

```

<?xml version="1.0" encoding="UTF-8"?>
<murmur>
    <simulator>
        <!-- Activate or not simulation -->
        <enable>1</enable>

        <!-- Sound input for simulators -->
        <soundInput>
            <device><!-->
                <nbChannels>1</nbChannels>
            </device>
        </soundInput>

        <!-- Devices -->
        <devices>
            <device>
                <id>deviceEchoSimulator01</id>
                <nblEDs>158</nblEDs>
                <length>4.0</length>
            </device>

            <!-->
            <device>
                <id>deviceEchoSimulator02</id>
                <nblEDs>158</nblEDs>
                <length>2.0</length>
            </device>
        </devices>
    </simulator>

    <!-- MadMapper -->
    <madmapper>stereolux.mad</madmapper>

    <!-- Raspberry -->
    <launchDevices>
        <ip>10.23.108.139</ip>
    </launchDevices>

```

3. Commande permettant d'indiquer le périphérique de captation sonore (micro intégré, carte son) et la ses spécificités (mono, stéréo). MURMUR a été développé pour une captation en mono. Il faut préciser également que en mode simulation, on peut représenter plusieurs devices avec 1 seul micro (voir page 5).

<device>0</device>: périphérique par défaut
<nbChannels>1</nbChannels>: mono

<device>1</device>: autre périphérique
<nbChannels>2</nbChannels>: stéréo

```

<murmur>
  <simulator>
    <!-- Activate or not simulation -->
    <enable>1</enable>
    <!-- Sound input for simulators -->
    <soundInput>
      <device><!-->
        <nbChannels>1</nbChannels>
      </soundInput>
    <!-- Devices -->
    <devices>
      <device>
        <id>deviceEchoSimulator01</id>
        <nbLEDs>158</nbLEDs>
        <length>4.0</length>
      </device>
    <!-->
    </devices>
  </simulator>
  <!-- MadMapper -->
  <madmapper>stereolux.mad</madmapper>
  <!-- Raspberry -->
  <launchDevices>
    <ip>10.23.108.139</ip>
  </launchDevices>
</murmur>

```

4. Commande permettant de définir le nombre de devices (chambre d'écho), la taille du ledstrip (ruban de LEDs) et le nombre de LEDs qui le composent. (voir page 5).

<id>deviceEchoSimulator01</id> : nom du device
<nbLEDs>158</nbLEDs> : 158 LEDs
<length>4.0</length> : longueur en m. du ledstrip

Pour configurer un 2e device ajouter en dessous :

```

<device>
<id>deviceEchoSimulator02</id>
<nbLEDs>158</nbLEDs>
<length>4.0</length>
</device>

```

```

<murmur>
  <simulator>
    <!-- Activate or not simulation -->
    <enable>1</enable>
    <!-- Sound input for simulators -->
    <soundInput>
      <device><!-->
        <nbChannels>1</nbChannels>
      </soundInput>
    <!-- Devices -->
    <devices>
      <device>
        <id>deviceEchoSimulator01</id>
        <nbLEDs>158</nbLEDs>
        <length>4.0</length>
      </device>
    <!-->
    </devices>
  </simulator>
  <!-- MadMapper -->
  <madmapper>stereolux.mad</madmapper>
  <!-- Raspberry -->
  <launchDevices>
    <ip>10.23.108.139</ip>
  </launchDevices>
</murmur>

```

5. Commande permettant de charger une composition madMapper enregistrée préalablement dans le fichier :

/data/Config/madmapper

```

<murmur>
  <simulator>
    <!-- Activate or not simulation -->
    <enable>1</enable>
    <!-- Sound input for simulators -->
    <soundInput>
      <device><!-->
        <nbChannels>1</nbChannels>
      </soundInput>
    <!-- Devices -->
    <devices>
      <device>
        <id>deviceEchoSimulator01</id>
        <nbLEDs>158</nbLEDs>
        <length>4.0</length>
      </device>
    <!-->
    </devices>
  </simulator>
  <!-- MadMapper -->
  <madmapper>stereolux.mad</madmapper>
  <!-- Raspberry -->
  <launchDevices>
    <ip>10.23.108.139</ip>
  </launchDevices>
</murmur>

```

6. Commande permettant le lancement de la Raspberry pi et donc de la captation sonore. Il est possible de lancer plusieurs Raspberry pi lorsque il y a plusieurs chambres d'écho. (voir page 5).

<launchDevices>
<ip>10.23.108.139</ip> : adresse ip de la Raspberry pi
</lauchDevices>

```

1 <scene>
2   <silhouettes>
3     <silhouette device="deviceEchoSimulator01">
4       <position x="0.0" y="0.0" z="4.0"/>
5     </silhouette>
6   </silhouettes>
7 </scene>

```

Line 1, Column 1 Tab Size: 4 XML

7. Si plusieurs devices sont activés (voir étape **4.**) il faut alors configurer le fichier **scene.xml** afin d'ajouter une silhouette dans l'interface et ainsi simuler plusieurs utilisateurs.
[/data/Config/scene.xml](#)

```

1 <scene>
2   <silhouettes>
3     <silhouette device="deviceEchoSimulator01">
4       <position x="2.0" y="0.0" z="4.0"/>
5     </silhouette>
6   </silhouettes>
7
8   <silhouettes>
9     <silhouette device="deviceEchoSimulator02">
10      <position x="-2.0" y="0.0" z="4.0"/>
11    </silhouette>
12  </silhouettes>
13
14 </scene>

```

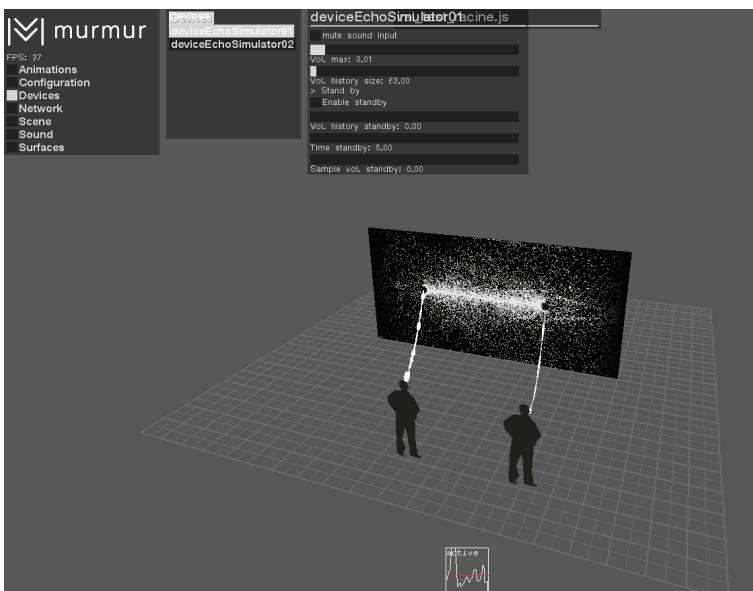
Line 5, Column 27 Tab Size: 4 XML

8. Afin d'ajouter une nouvelle silhouette dans la scène il faut taper la commande:

```

<silhouettes>
<silhouette device=>deviceEchoSimulator02<>:nom de la deuxième silhouette
<position x=>-2.0<>y=>0.0<>z=>4.0</>:position de la silhouette
</silhouette>
</silhouettes>

```



9. Par défaut et lorsqu'il y a qu'un seul device la silhouette ce place au centre de la scène :

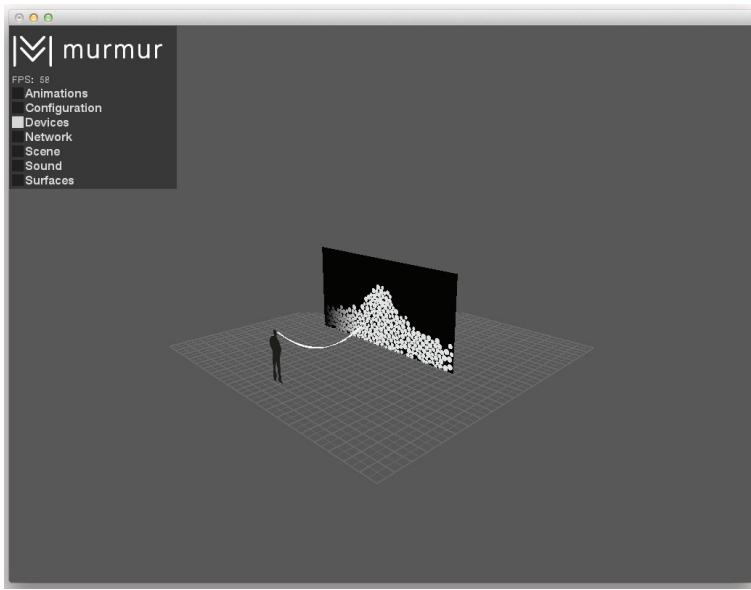
```

<silhouette device=>deviceEchoSimulator01<>:nom de la silhouette n°1
<position x=>-0.0<>y=>0.0<>z=>0.0</>:position de la silhouette au centre de la scène

```

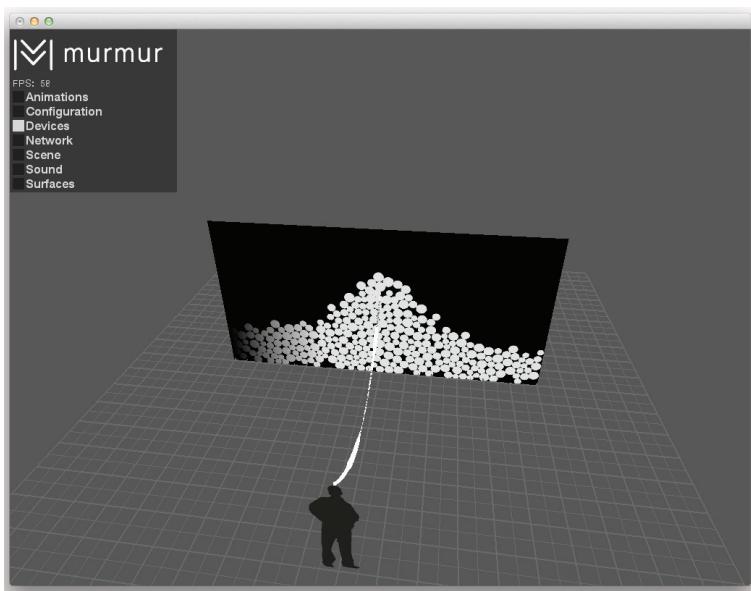
|☰| prise en main de l'interface

Le logiciel MURMUR est représenté sous la forme d'un environnement 3D permettant de calibrer l'espace physique (l'espace de projection) mais également les différentes interactions possibles accessibles par un système d'onglets.

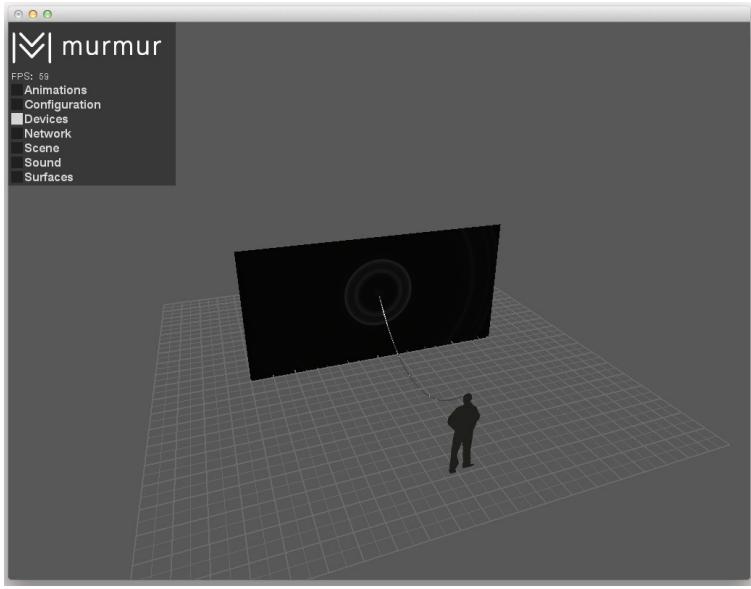


10. Interface générale de l'application MURMUR.
La scène 3d représente 3 éléments :

- l'utilisateur
- le ledstrip
- la surface de projection



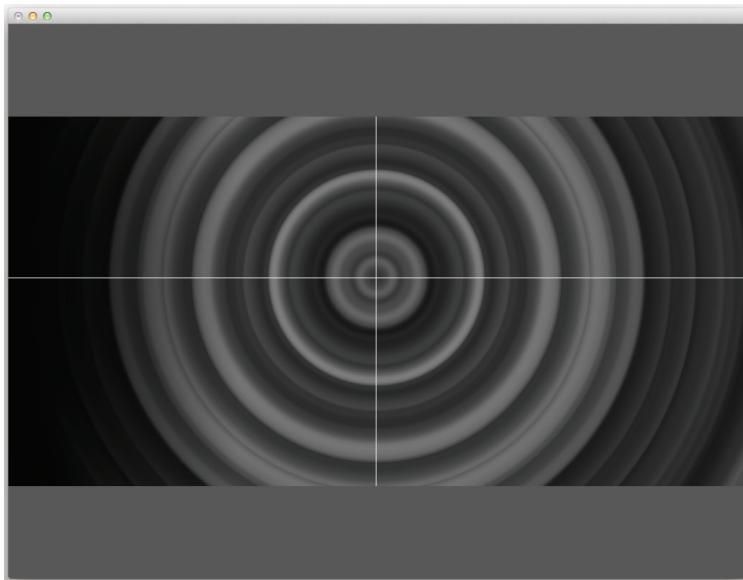
11. L'interface se compose d'une scène 3D et d'un ensemble d'onglets situés sur la gauche, que l'on active en cliquant dessus. Ces onglets contiennent les différents outils permettant d'ajuster et de calibrer MURMUR.



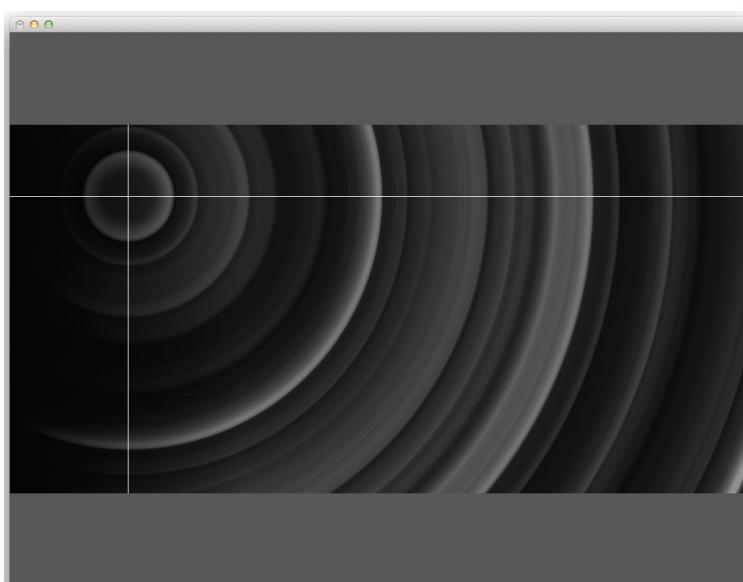
12. Il est possible de se déplacer dans la scène en utilisant la souris.

|▽| écran de projection

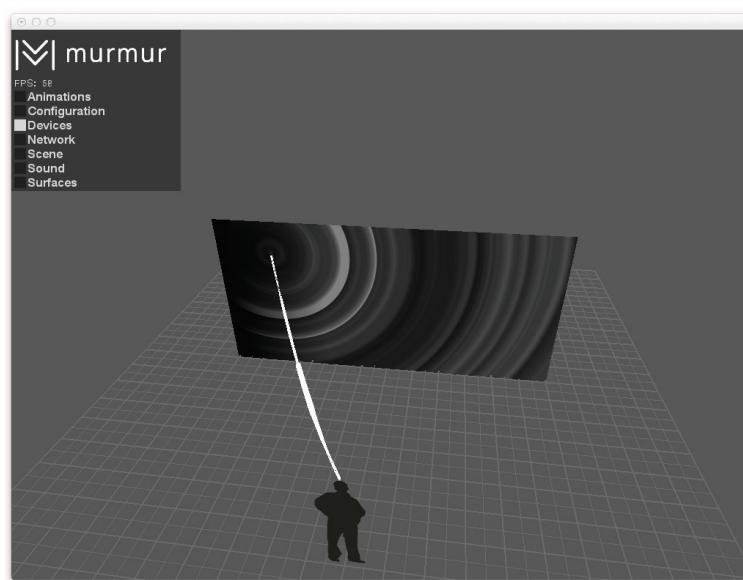
L'écran de projection permet de visualiser la projection et les animations de manière frontal et de faire abstraction de l'environnement 3D.



13. Il est possible de passer l'écran de projection en fullscreen a l'aide de la barre d'espace.



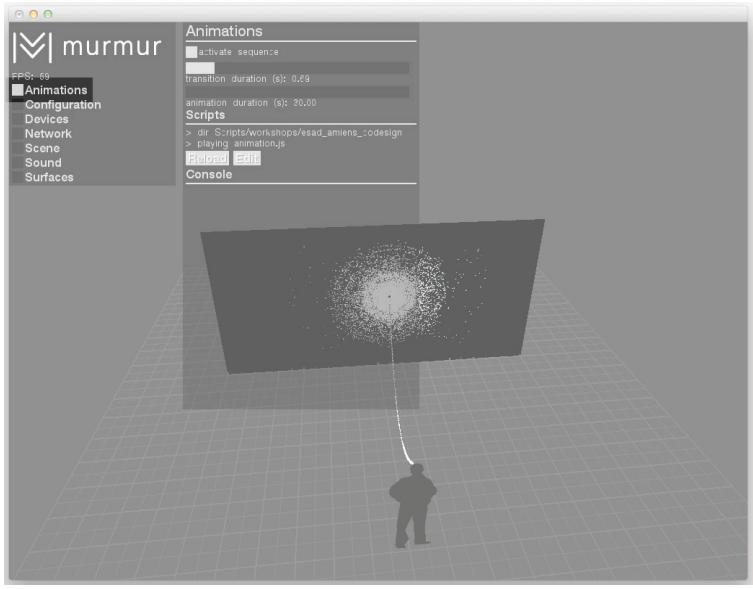
14. À partir de cet écran il est possible de changer le point d'accroche du ledstrip. Sa position est indiquée par une croix qui par défaut est centrale. Pour la modifier il suffit de cliquer à l'endroit voulu. Pour voir la position du device il ne faut pas oublier de désactiver l'option “quadwarping” dans l'option “Surfaces”.



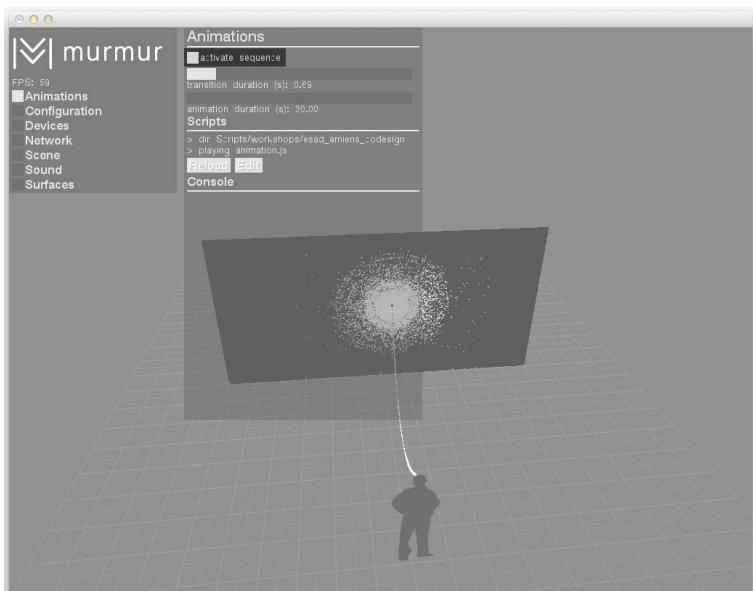
15. Pour revenir à l'écran principal il suffit d'utiliser à nouveau la barre d'espace. On remarque que l'accroche du ledstrip a bien été déplacée.

|▽| animation

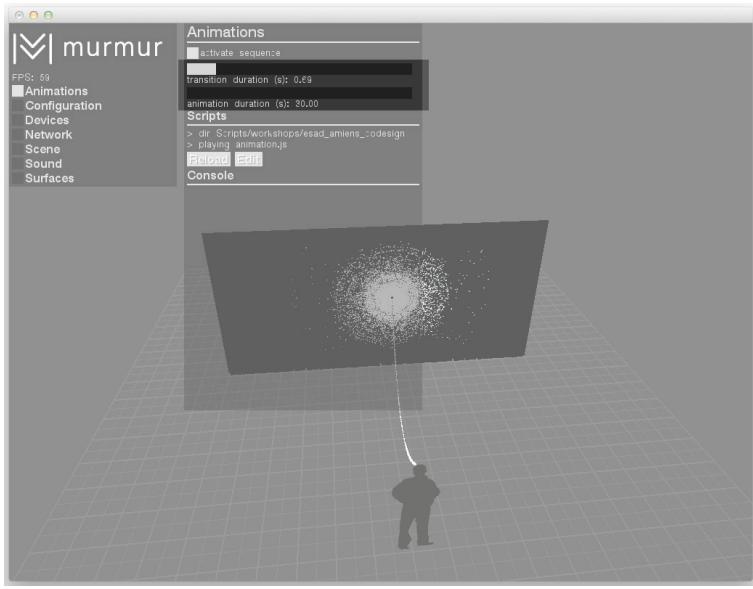
L'onglet animation permet de calibrer les différentes animations. Il existe deux sortes d'animation, celles rédigées en Cpp. qui font partie intégrante du logiciel et non modifiables via un éditeur de texte et les animation Js. qui sont gérées dynamiquement, que l'on peut modifier et observer en temps réel. Il existe également deux sorte de visualisations d'animations. Le mode manuel qui permet de changer d'animation manuellement et le mode séquence qui lance une playlist d'animations réglées au préalable.



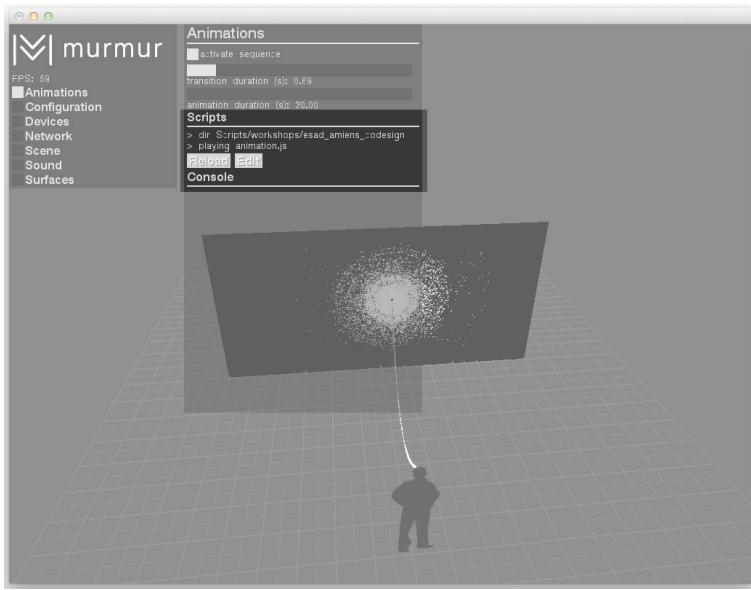
16. L'onglet “Animation” permet de contrôler les différents visuels et animations. Il faut préciser que tout réglage effectué dans le soft est sauvegarder automatiquement à la fermeture du logiciel.



17. “Activate sequence” permet de lancer une suite d’animation Js ou Cpp à la manière d’une liste de lecture. Certaines animation peuvent être modifiées directement à partir du soft, dans ce cas là un nouvel onglet avec différents boutons de réglage apparaît.



18. “Transition duration” permet de créer un fondu plus ou moins long (en seconde) entre les différentes animations tandis que “animation duration” permet de régler la durée de lecture des animations. Le temps minimum d'une animation est de 30 secondes.



19. La partie “Scripts” permet de visualiser le nom de l’animation et sa localisation dans l’arborescence. Par défaut les scripts Js se trouve dans le dossier :
[/data/Scripts/](#)

Le bouton “Edit” permet d’ouvrir dans l’éditeur de texte l’animation en cours de lecture pour la modifier. Une fois modifiée utiliser le bouton “reload” pour recharger l’animation. (cf étape **22**).

surfacemain.xml

```

1 <surface id="main">
2   <animations type="cpp">
3     <animation>shaderWave</animation>
4     <animation>particleMega2</animation>
5     <animation>box2DCircles</animation>
6   </animations>
7
8   <animations type="js" folder="workshops/esad_amiens_codesign">
9     <!--
10    <animations type="js" folder="murmur">
11    <animations type="js" folder="workshops/stereolux_jcc2">
12    -->
13   </animations>
14
15   <timeline>
16     <animation>tunnel</animation>
17     <animation>particleMega2</animation>
18     <animation>box2DCircles</animation>
19     <animation>shaderWave</animation>
20     <animation>waves2.js</animation>
21   </timeline>
22
23 </surface>
24
25

```

Line 1, Column 1 Spaces: 4 XML

20. Le fichier **surfacemain.xml** permet de configurer les séquences d’animations :
[/data/Config/surfaces/surfacemain.xml](#)

surfacemain.xml

```

1 <surface id="main">
2   <animations type="cpp">
3     <animation>shaderWave</animation>
4     <animation>particleMega2</animation>
5     <animation>box2DCircles</animation>
6   </animations>
7
8   <animations type="js" folder="workshops/esad_amiens_codesign">
9     <!--
10    <animations type="js" folder="murmur">
11    <animations type="js" folder="workshops/stereolux_jcc2">
12    -->
13   </animations>
14
15   <timeline>
16     <animation>tunnel</animation>
17     <animation>particleMega2</animation>
18     <animation>box2DCircles</animation>
19     <animation>shaderWave</animation>
20     <animation>waves2.js</animation>
21   </timeline>
22
23 </surface>
24
25

```

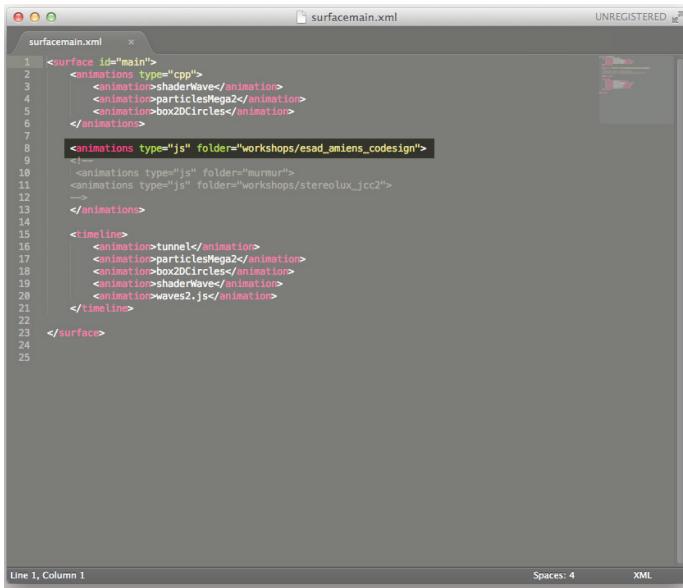
Line 1, Column 1 Spaces: 4 XML

21. Pour déclarer les animations Cpp il suffit de modifier les lignes de suivante en ajoutant les noms des animations :

```

<animations Type="cpp">
<animation>chaderWave</animation>:anim. 1
<animation>particleMega2</animation>:anim. 2
<animation>box2DCircles</animation>:anim. 3
</animations>

```

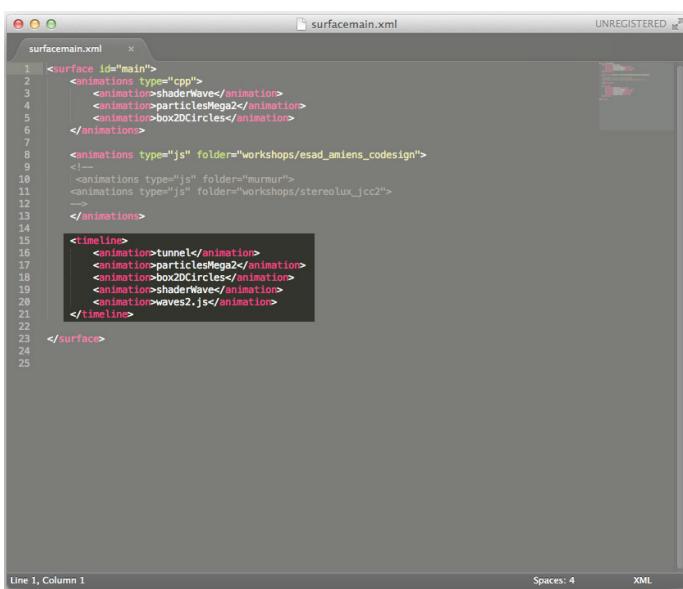


```
1 <surface id="main">
2   <animations type="cpp">
3     <animation>shaderWave</animation>
4     <animation>particlesMega2</animation>
5     <animation>box2DCircles</animation>
6   </animations>
7
8   <animations type="js" folder="workshops/esad_amiens_codesign">
9     <!--
10    <animations type="js" folder="murmur">
11    <animation>particlesMega2</animation>
12    <!-->
13  </animations>
14
15  <timeline>
16    <animation>tunnel</animation>
17    <animation>particlesMega2</animation>
18    <animation>box2DCircles</animation>
19    <animation>shaderWave</animation>
20    <animation>waves2.js</animation>
21  </timeline>
22
23 </surface>
24
25
```

22. La commande suivante permet de spécifier un sous dossier contenant les animations Js dans le dossier :

/data/Scripts/

<animations type="js" folder="murmur">



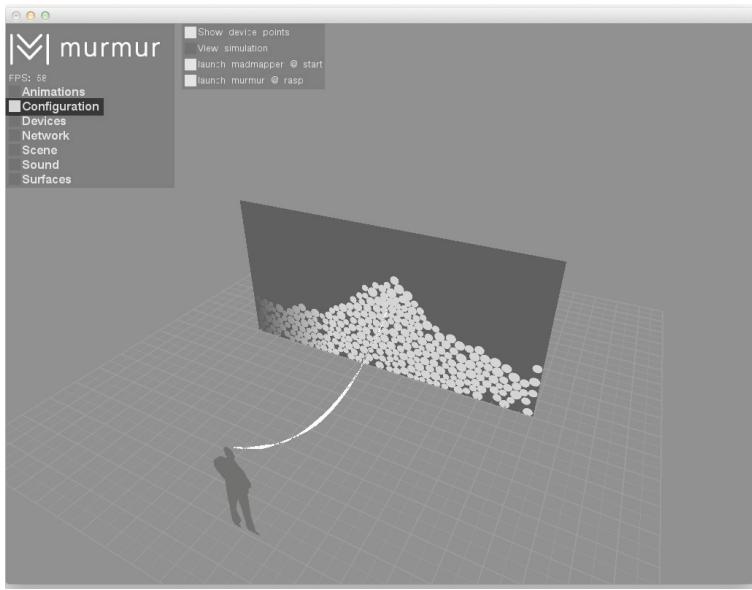
```
1 <surface id="main">
2   <animations type="cpp">
3     <animation>shaderWave</animation>
4     <animation>particlesMega2</animation>
5     <animation>box2DCircles</animation>
6   </animations>
7
8   <animations type="js" folder="workshops/esad_amiens_codesign">
9     <!--
10    <animations type="js" folder="murmur">
11    <animation>particlesMega2</animation>
12    <!-->
13  </animations>
14
15  <timeline>
16    <animation>tunnel</animation>
17    <animation>particlesMega2</animation>
18    <animation>box2DCircles</animation>
19    <animation>shaderWave</animation>
20    <animation>waves2.js</animation>
21  </timeline>
22
23 </surface>
24
25
```

23. La commande suivante permet de créer la séquence des animations Js et Cpp (cf étape 17) :

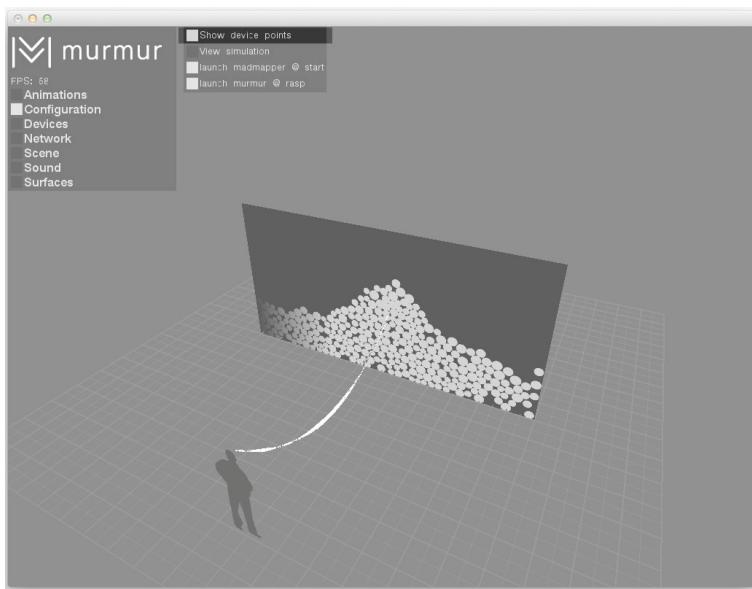
```
<timeline>
<animation>tunnel</animation>:anim. 1
<animation>particlesMega2</animation>:anim. 2
<animation>box2DCircles</animation>:anim. 3
<animation>shaderWave</animation>:anim. 4
<animation>waves2.js</animation>:anim. 5
</timeline>
```

|ℳ| configuration

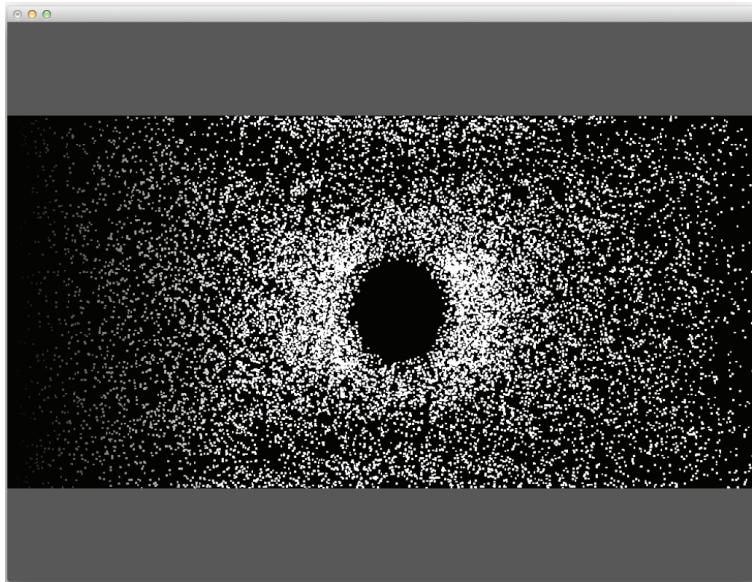
L'onglet configuration est particulièrement important en mode installation. Il permet de régler certaines options et de faire le lien avec d'autres logiciels comme madMapper.



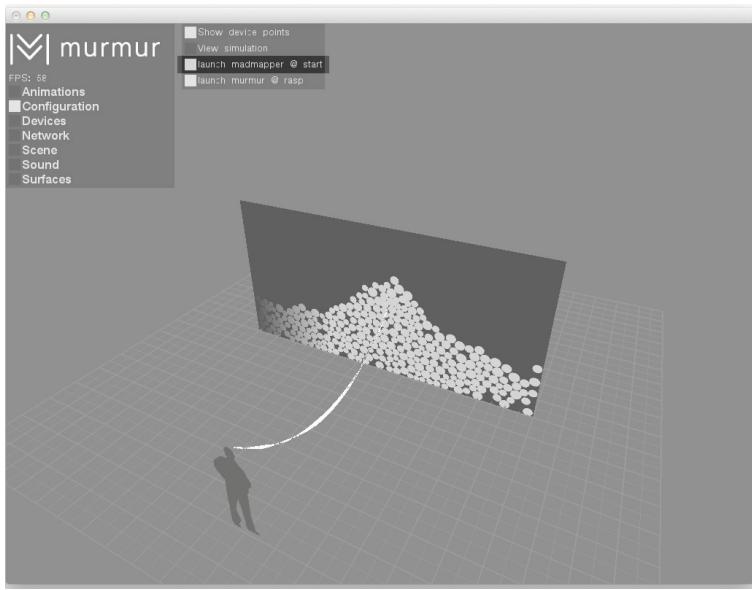
24. L'onglet “Configuration” permet de régler certains paramètres lors de l’ouverture de MURMUR mais également d’activer certaines options du soft.



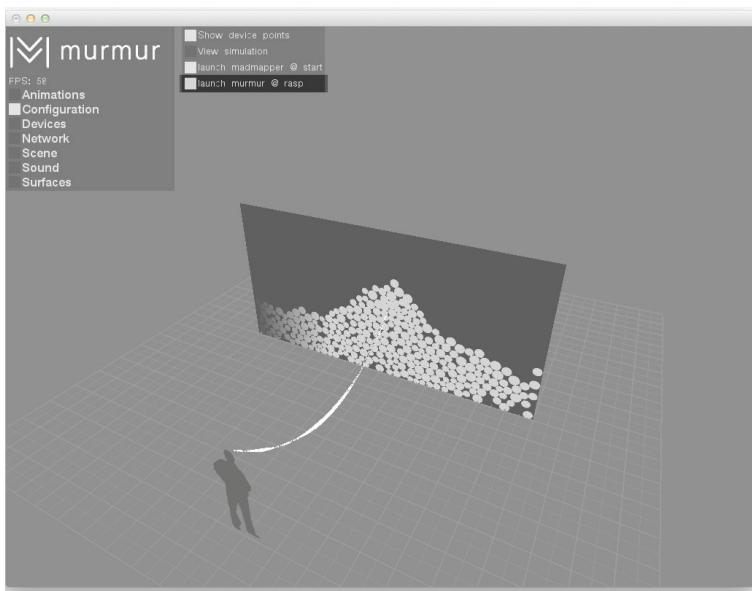
25. “Show device points” permet d'afficher ou de masquer la position du device lorsque l'on utilise le plein écran (barre d'espace). Pour voir la position du device il ne faut pas oublier de désactiver l'option “quadwarping” dans l'option “Surfaces” (cf étape **53**).



26. Le centre du device n'est plus visible une fois l'option “Show device points” désactivée.



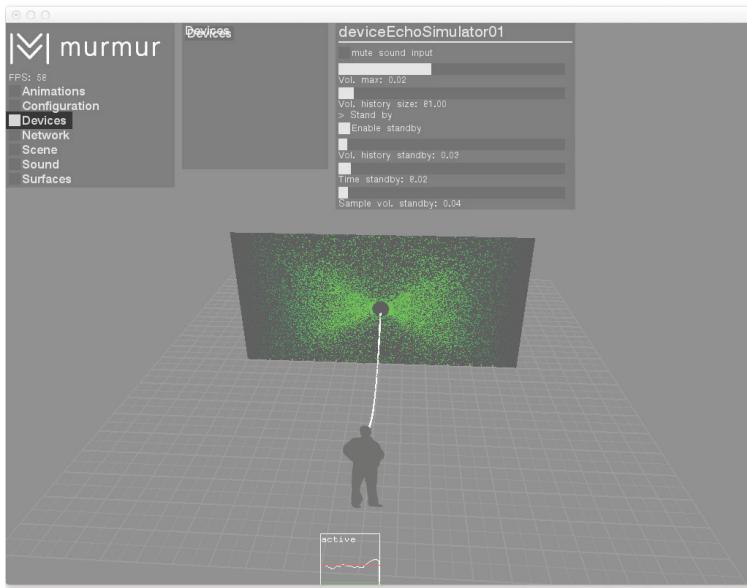
27. L'option “launch madmapper @ start” permet lors du prochain lancement de MURMUR en mode installation d'ouvrir automatiquement madMapper et de charger une composition indiquée lors de l'étape **5**. L'ouverture de madMapper n'est pas forcément systématique, la plupart des réglages de mapping peuvent s'effectuer avec l'option “quadwarping” (cf étape **53**).



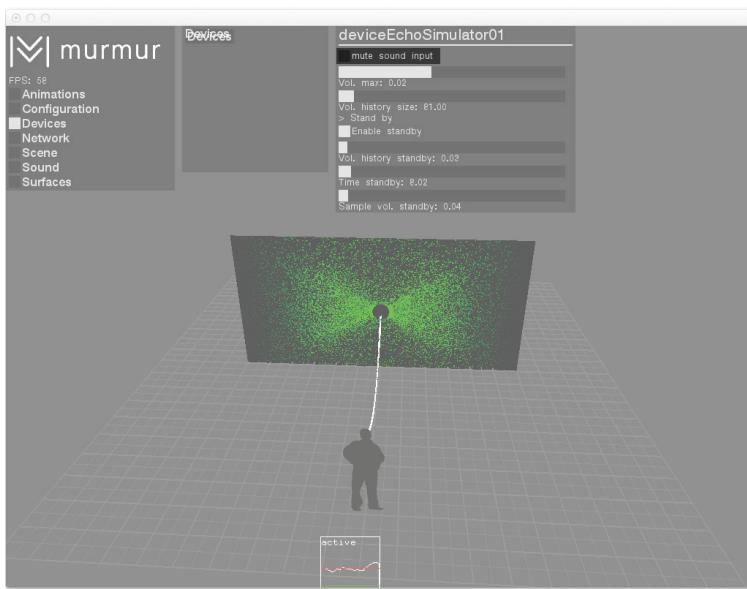
28. L'option “launch murmur @ rasp” permet lors de la prochaine ouverture de MURMUR en mode installation de lancer automatiquement la ou les Raspberry pi (cf étape **6**).

|ℳ| devices

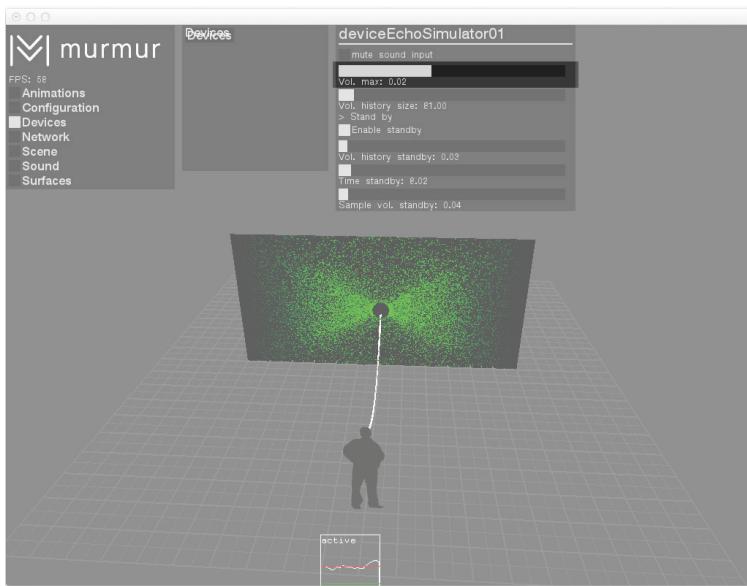
Avec MURMUR on utilise le terme devices pour parler des chambres d'écho. Ces devices sont composés d'une Raspberry pi possédant son propre nom sous la forme d'une adresse IP, ainsi chaque devices a une identité propre. Il est possible d'ajouter autant de chambres d'écho que l'on souhaite (cf étape **4**).



29. L'onglet “devices” permet de calibrer la captation sonore avec ou sans utilisateurs (mode veille cf étape **33**).

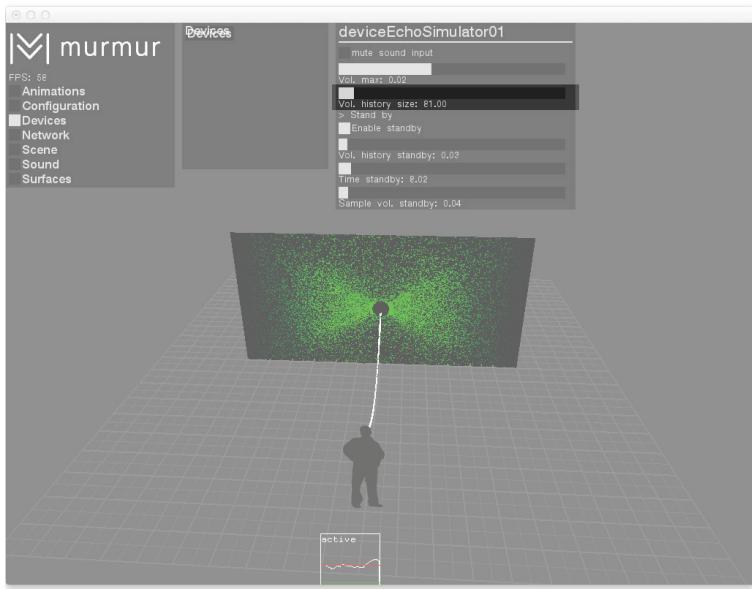


30. En activant “mute sound input” vous coupez toute captation sonore, l’application tourne toujours, mais aucune information sonore n’est récupérée.



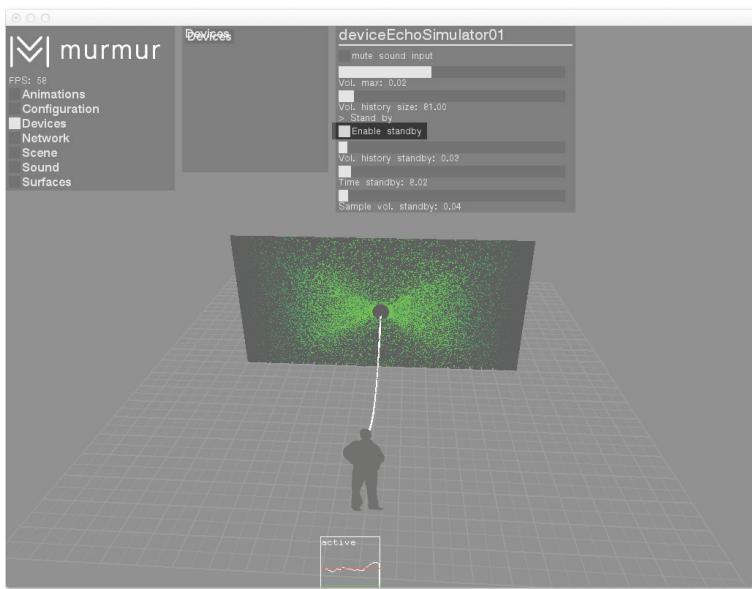
31. “Vol. max” permet d’ajuster le volume du signal sonore capté. Les valeurs suivantes peuvent changer en fonction des situations.

“Vol. max: 0.00” le signal capté est au maximum
“Vol. max: 0.04” le signal capté est au minimum



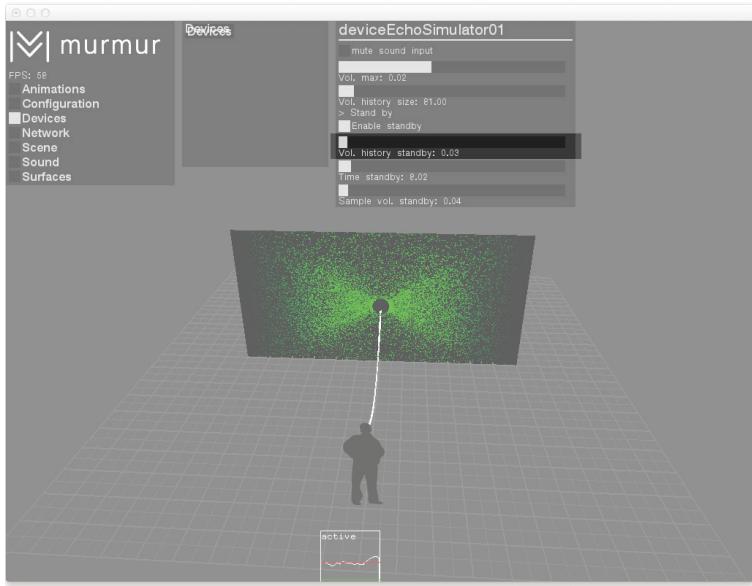
32. "Vol. history sizes" intervient sur la vitesse de déplacement du signal le long du ledstrip. Les valeurs suivantes peuvent changer en fonction des situations.

"Vol. history sizes: 500.00" le signal est lent
"Vol. history sizes: 50.00" le signal est rapide



33. "Enable standby" active ou désactive le mode standby.

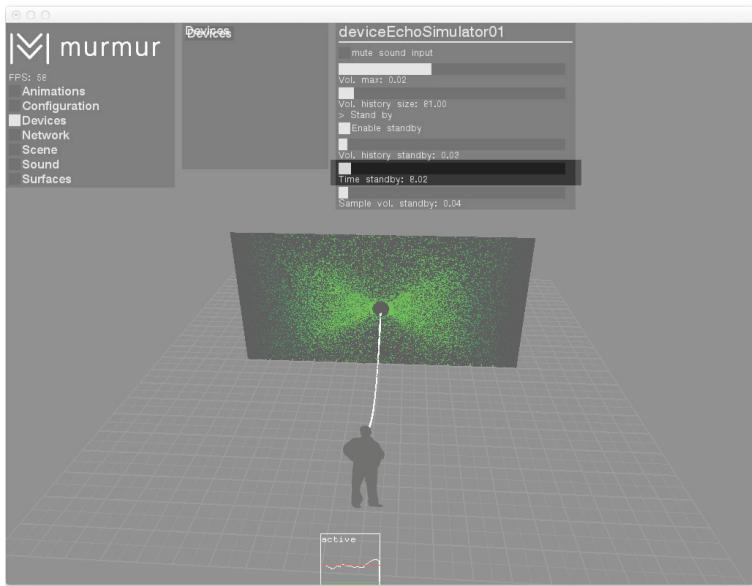
Le mode se lance lorsque aucun son n'est capté, dans ce cas MURMUR va automatiquement lancer un court signal sonore à intervalle régulier afin que l'installation soit visuellement active.



34. "Vol. history standby" permet d'ajuster le niveau sonore minimum sous lequel l'application passe en mode standby (cf étape **37**).

"Vol. history standby: 0.00" le mode standby se lance lorsque le signal sonore est à 0

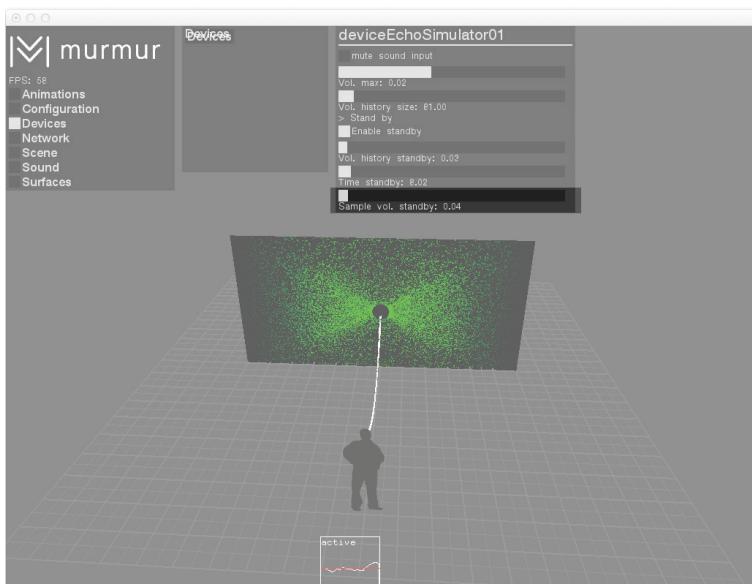
"Vol. history standby: 0.75" le mode standby se lance lorsque le signal sonore est en dessous de 0.75



35. "Time standby" règle le temps d'attente, lorsque le niveau sonore minimum est atteint pour lancer l'impulsion sonore du mode standby.

"Time standby: 5.00" 5 secondes avant impulsion

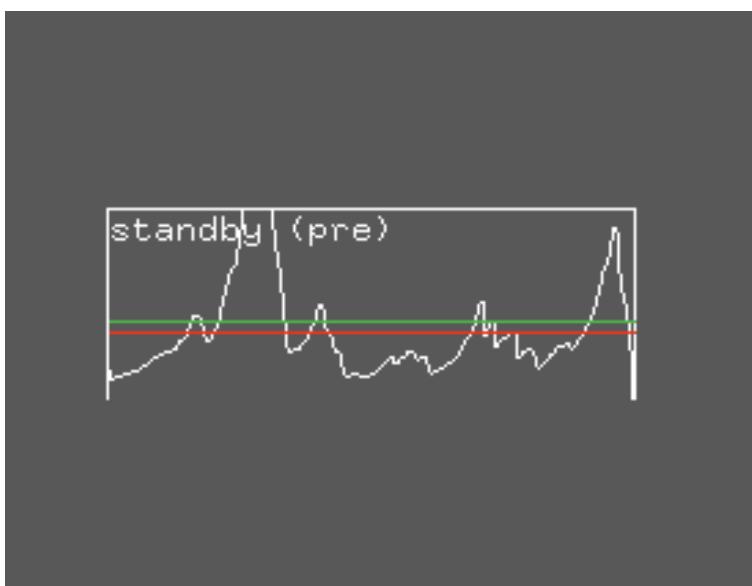
"Time standby: 60.00" 1 minute avant impulsion



36. "Sample vol. standby" ajuste le volume de l'impulsion sonore.

"Sample vol. standby: 0.00" volume au plus bas

"Sample vol. standby: 1.00" volume au plus haut



37. La fenêtre située au bas de l'écran retranscrit visuellement les changements et les ajustements effectués lors des étapes :

31, 32, 33, 34, 35 et 36.

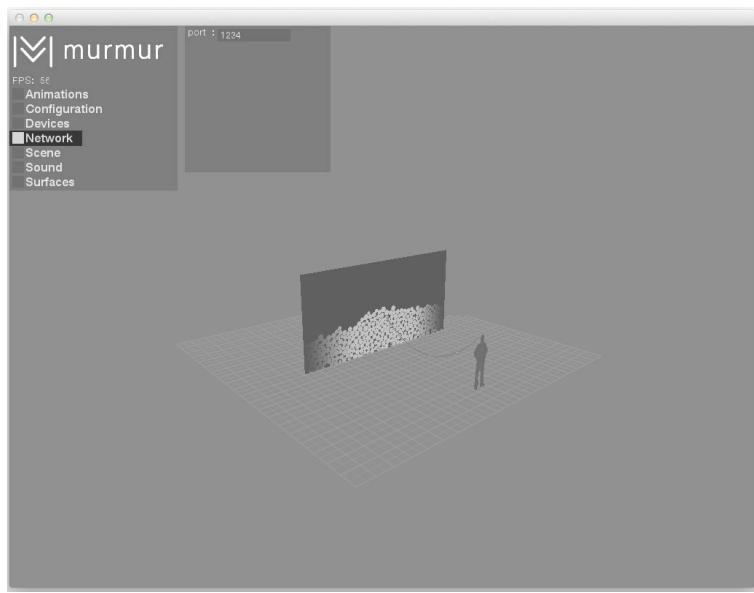
La courbe blanche représente la captation sonore, la ligne rouge représente le volume moyen et la ligne verte est assignée au volume minimum avant le lancement du mode standby. Cette fenêtre indique également le mode en cours de lecture :

"active" déroulement classique

"standby pre" temps avant le lancement du mode standby (cf étape 35)

"standby" mode standby

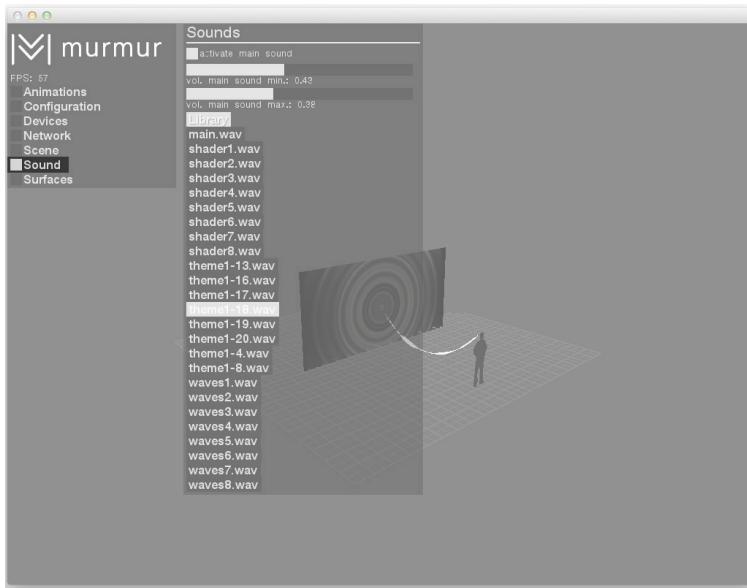
|ℳ| network



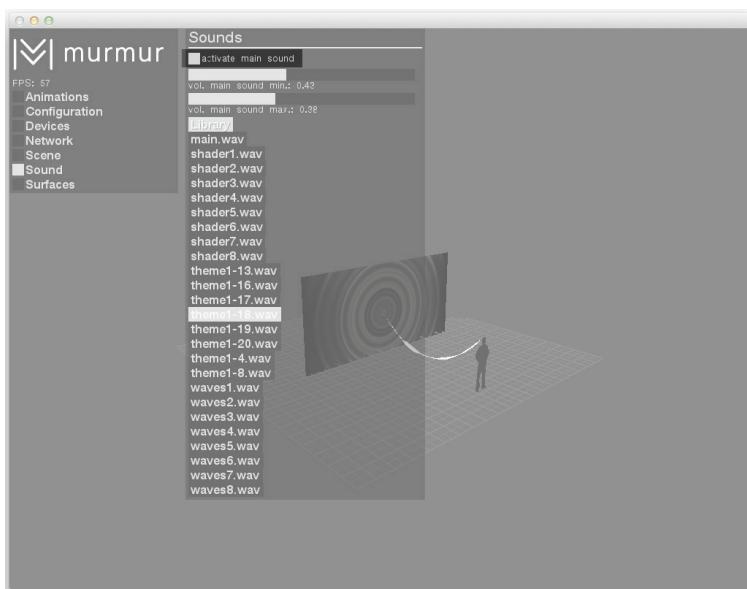
38. L'onglet “Network” permet d’indiquer le port pour l’échange de messages osc. Par défaut MURMUR utilise le port 1234 (cf page 5).

|ℳ| sound

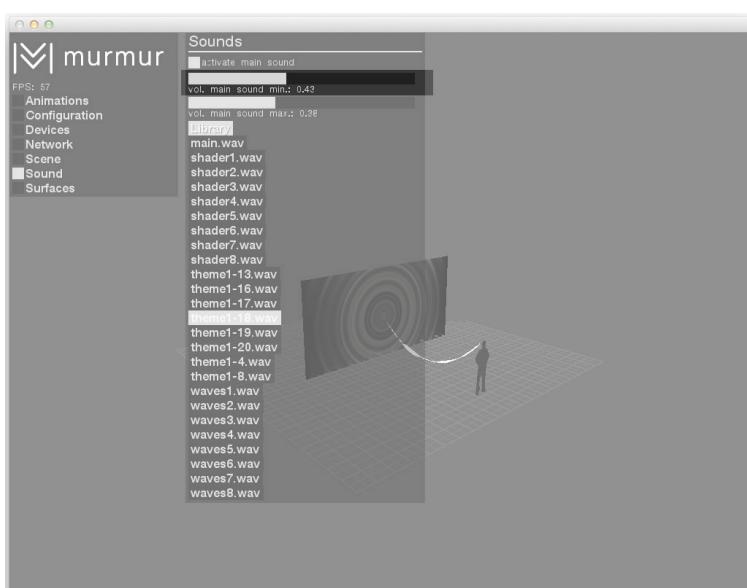
L'onglet sound rassemble les différents outils permettant le calibrage du son inhérent au logiciel, il permet en autre de visualiser la bibliothèque regroupant l'ensemble des samples de MURMUR.



39. L'onglet "Sound" permet de calibrer les sons internes à l'application MURMUR.

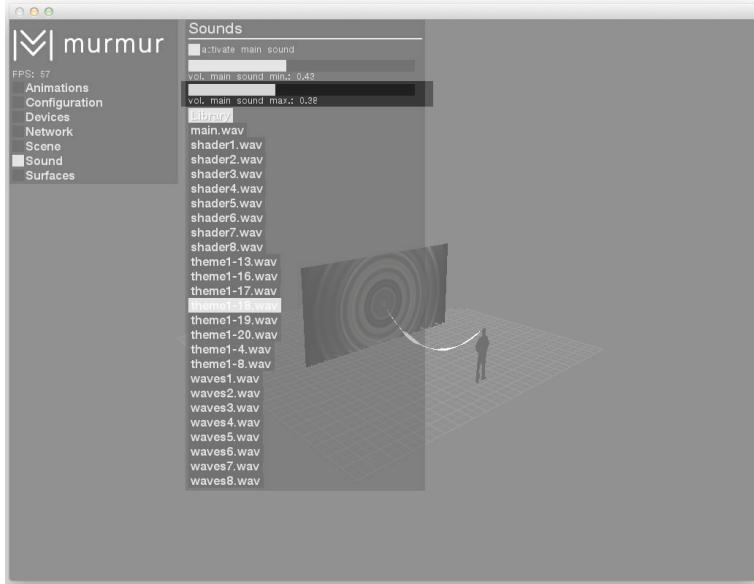


40. "activate main sound" active ou désactive les sons ambiants de l'application MURMUR.

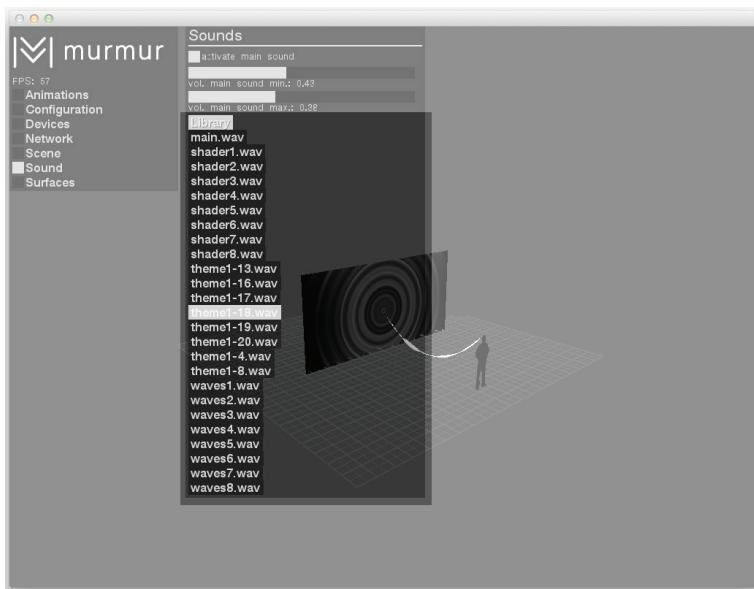


41. "Vol. main sound min" régle le volume sonore du son ambiant.

42.

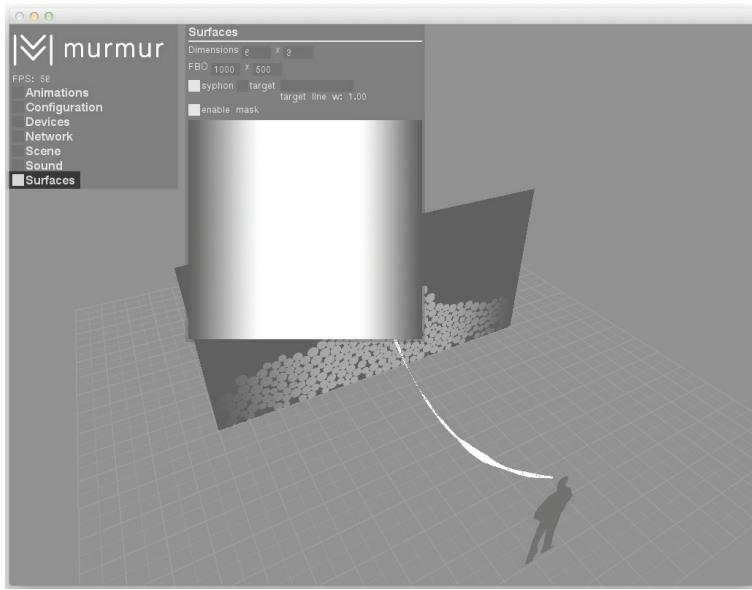


43. "Library" regroupe tous les samples de l'installation MURMUR. Ils sont présents dans le dossier:
</data/Sounds/>

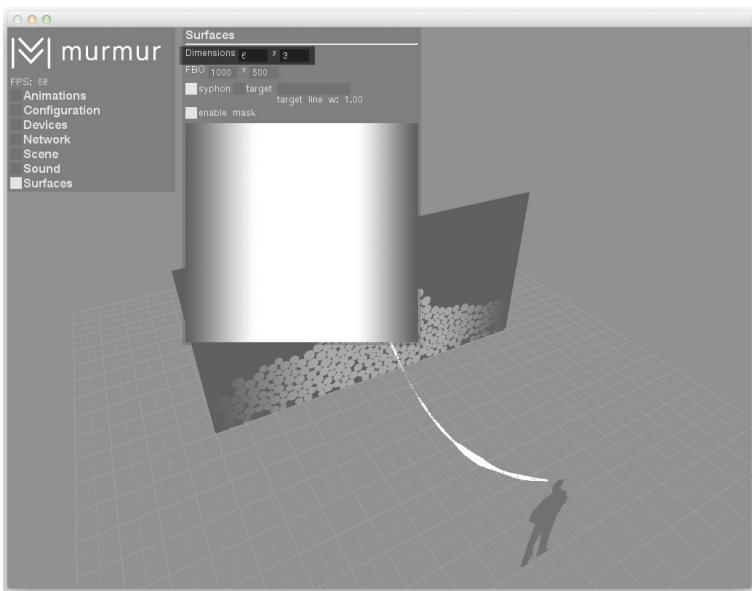


|▽| surface

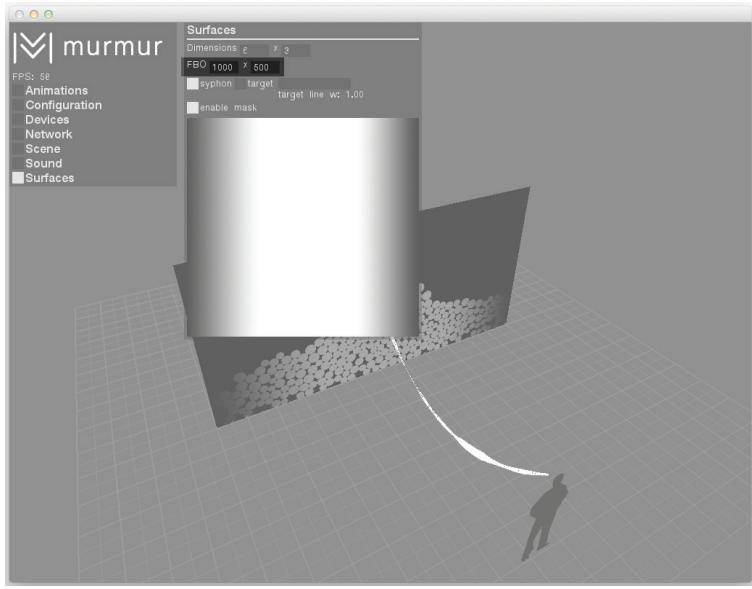
L'onglet surface permet de gérer les options liées à la surface de projection et à la projection elle même.



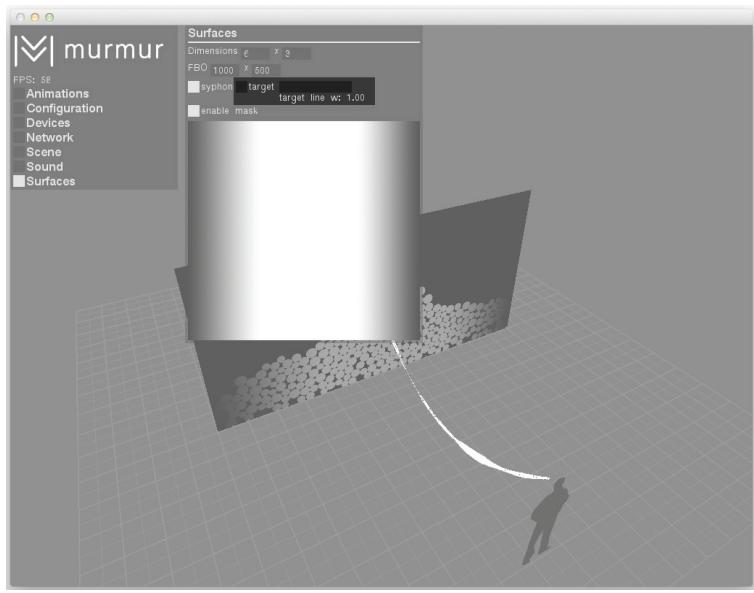
44. L'onglet "Surfaces" contient les réglages concernant l'environnement de projection, sa taille et ses spécificités.



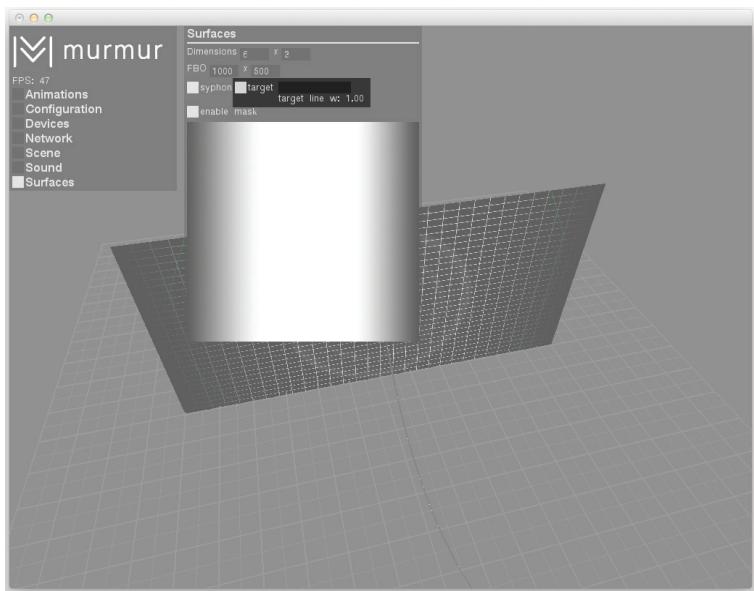
45. "Dimensions" permet de régler la taille de la surface de projection, sa hauteur et sa largeur en mètre.



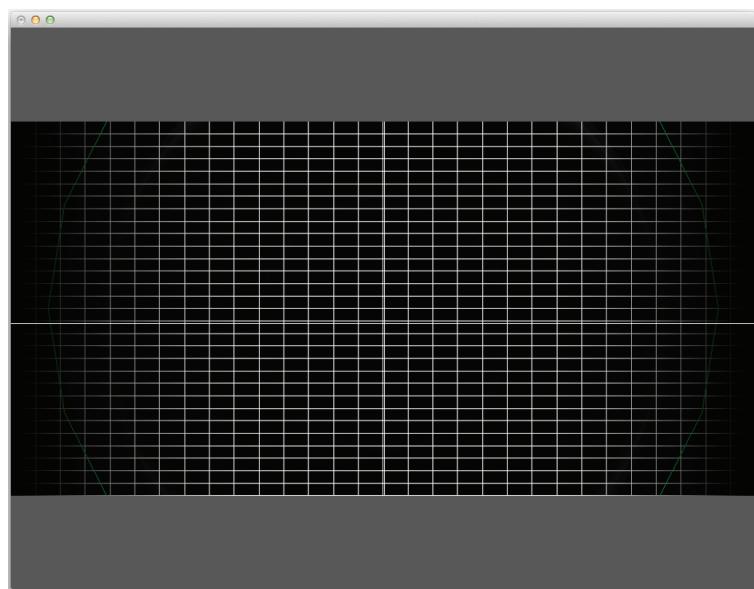
46. "FBO" calibre proportionnellement la surface où sera dessinées les animations, il faut régler ce paramètre de manière homothétique aux dimensions référencées lors de l'étape **45**.



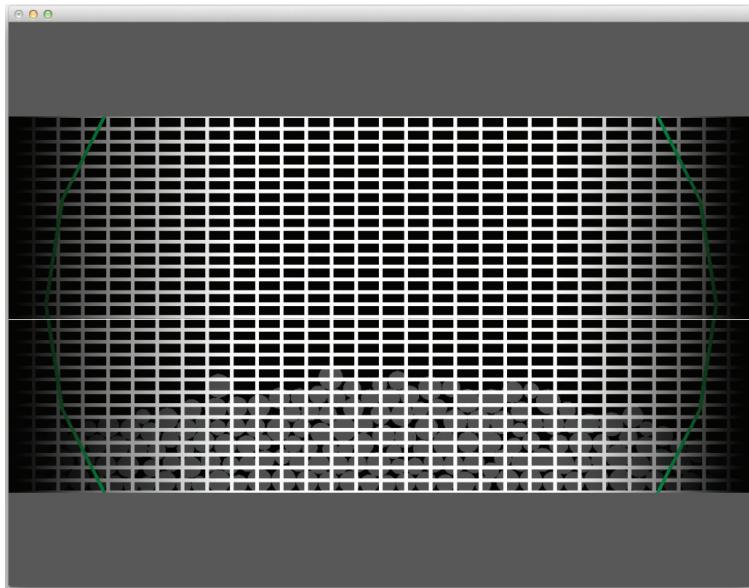
47. Le paramètre “Target” une fois activé affiche une grille sur la surface de projection permettant de faciliter le mapping via madMapper.



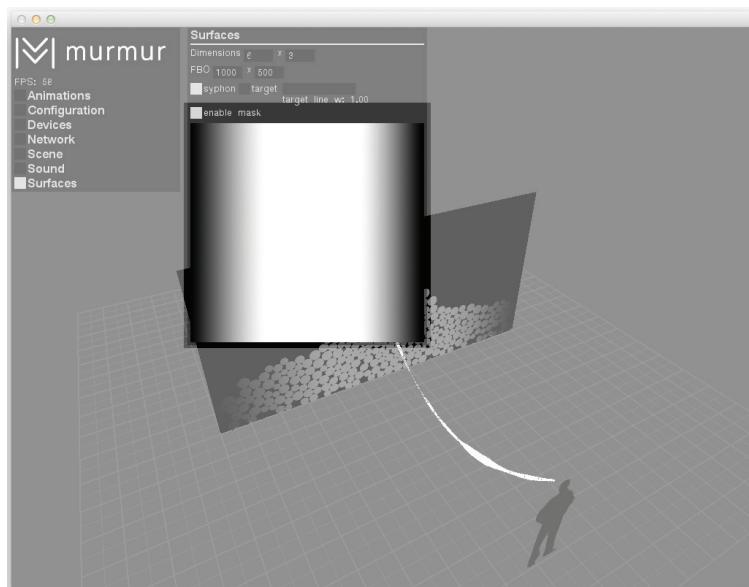
48. “Target line w” modifie l’épaisseur des lignes de la grille activée à l’aide du bouton “target”.



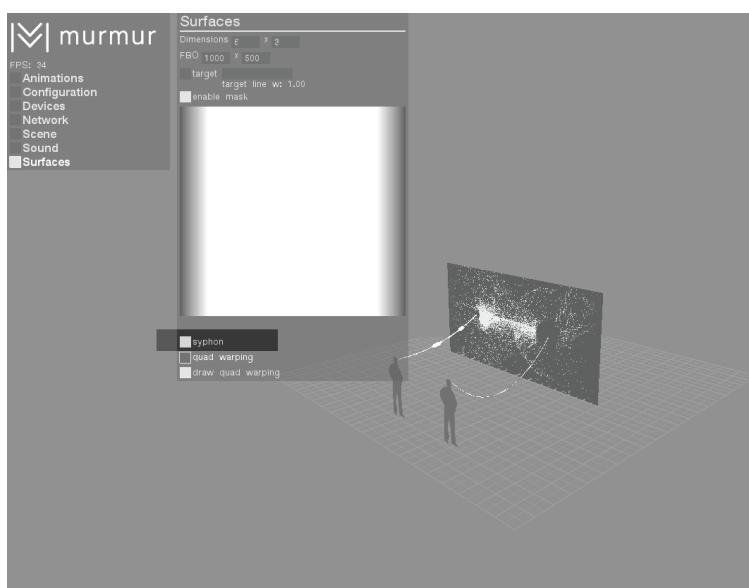
49. “Target line w: 1.00”.



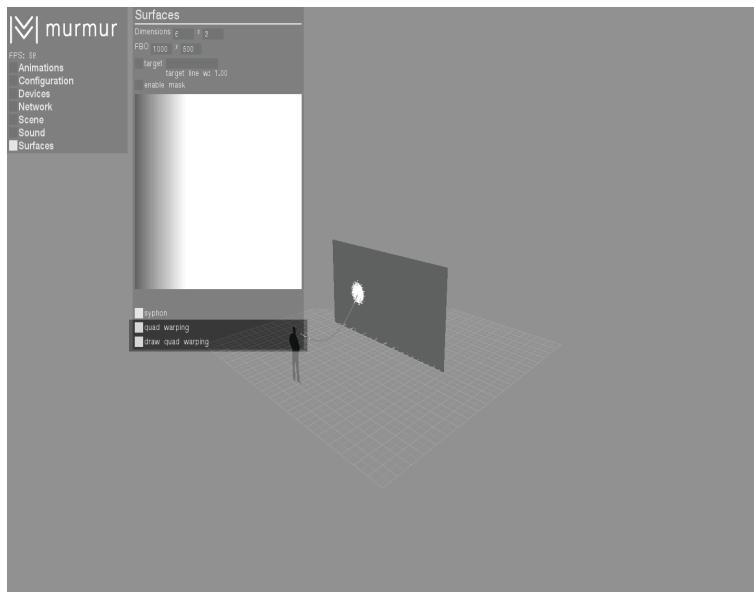
50. “Target line w:8.00”.



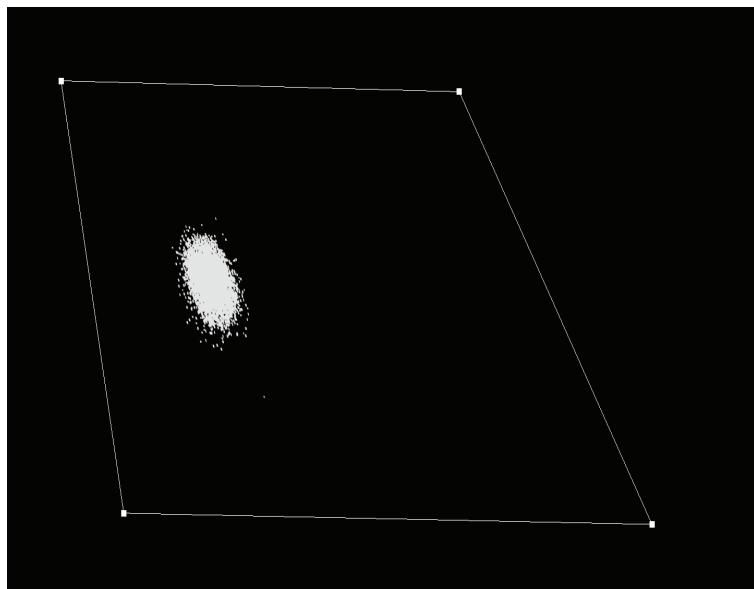
51. “enable mask” permet d’activer un masque qui viendra se superposer sur l’écran de projection, ainsi en ajoutant un dégradé il est possible de simuler des murs et un espace physique virtuel par un fondu au noir.
Pour ajouter un masque il suffit de réaliser une image carrée que l’on ajoutera dans l’application par un simple drag & drop.



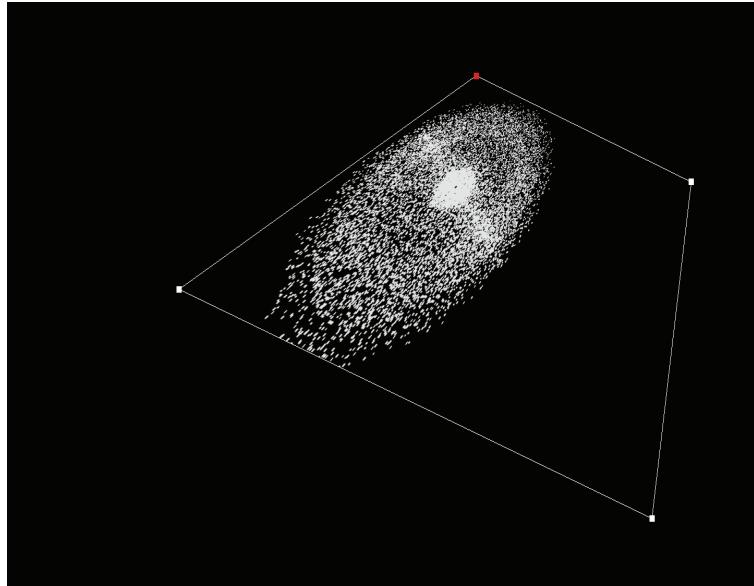
52. “syphon” permet d’activer le partage d’écran entre le soft et madMapper, ainsi le calibrage et le mapping sont plus faciles à réaliser. Bien sur il n’est pas forcement obligé de l’activer si les conditions de projection ne nécessitent pas l’aide du logiciel madMapper.



53. Les options “quad warping” et “draw quad warping” permettent d’effectuer une déformation de l’image afin de modifier la perspective et s’adapter à l’environnement de projection ou de contrebalancer un défaut lié au vidéoprojecteur. Si les défauts de projections sont trop importants, activer l’option “syphon” et utilisez madMapper (cf étape **52** et **5**).



54. Une fois les options “quad warping” et “draw quad warping” activées utilisez le raccourci clavier “barre d’espace” pour accéder à l’interface du “quad warping”.



55. Pour modifier le quadrilatère de projection, utilisez la souris pour déplacer les coins et obtenir la déformation souhaitée.