

Documentation Maison Sensible

05/08/15

Analyse du teaser

4 atmosphères à approfondir

1. **Paix** : battement du début du teaser 0-30" – le battement graphique devra être généré par Max, le battement audio, une respiration a été ajoutée (le sample audio doit être ajouté dans un player de Max et être déclenché de façon synchrone avec le battement graphique). Battement global mais irrégulier. Echantillon « polystirene » et « sample phase basse faible ».
Sur la paix seulement actuellement, le grattement fait vibrer le graphisme et produit des sons variables (2 familles de son pour les grattements en fonction du type de son de fond – paix1 vs paix2).
2. **Confiance** : 1' à 1'25". faire varier les textures d'accélération pour avoir des boucles plus ou moins longue. Echantillon « voiture gros gras ».
 1. **Moisissure** : 1'42-2'00. Avec une pulsation. Avoir un son indépendant qui envoie ces pulsations. (pourrait être intégré dans confiance)
 2. **Hiéroglyphe** : 2'01-2'20
3. **Mort de peur**: 2'49"-3'00. Saturation progressive vers le blanc – faire en sorte que les flashes fassent vibrer le blanc. L'absence d'agression fait retomber au noir.
Avoir une diffusion horizontale des particules.
4. **Peur** : graphisme 3'39"-> fin & audio 3'22"->3'39" + impact de lune à 3'45" (lors d'interactions sporadiques) qui permet de revenir à la paix. Echantillon « spatule » & « grognement ».
Avoir une ligne blanche en bas de l'écran qui remonte.
5. **Colère** à implémenter ?

Pour les transitions, faire un son fort spécifique pour masquer des cassures.

Avoir une transition graphique dont la durée est définissable dynamiquement.

Hierarchie des répertoires

répertoire de base

C:\Users\christian\Desktop\MaisonSensible

contient

- les scripts de lancement de la partie graphique en écran simple ou double maisSens.PG.bat et maisSens.PG-single.bat
- le script de lancement du patch puredata maisSens.PD.bat

répertoire des fichiers binaires

C:\Users\christian\Desktop\MaisonSensible\bin

contient

- les librairies de l'application graphique (dll)
- l'exécutable de la partie graphique Porphyrograph_maisSens.exe qui est régénéré à partir du projet Visual C++

répertoire du projet Visual C++ de Porphyrograph Porphyrograph.sln (rendu visuel)

C:\Users\christian\Desktop\MaisonSensible\Porphyrograph-VC

répertoire des fichiers source du projet Visual C++ de Porphyrograph (rendu visuel)

C:\Users\christian\Desktop\MaisonSensible\Porphyrograph-maisSens-src

répertoire des applications visuelles et audio

C:\Users\christian\Desktop\MaisonSensible\maisSens

répertoire de l'application Max/MSP

C:\Users\christian\Desktop\MaisonSensible\maisSens\maisSensMaxMSP

répertoire de l'application graphique Porphyrograph

C:\Users\christian\Desktop\MaisonSensible\maisSens\maisSensPorphyrograph

contient

- le fichier LYM_maisSens.pd de contrôle de l'application dans puredata
- **dans les sous-répertoires du répertoire blenderModels** : les modèles Blender de chacun des projets
- **dans le répertoire cmd** : le fichier de commande PG_configAndHeaderGenerator.prl pour régénérer le fichier de configuration à partir des fichiers de scénario et du header de fichier de configuration
- **dans le répertoire configuration** : les deux header de fichier de configuration LYM_maisSens-head.conf et LYM_maisSens-head-single.conf (pour écran double ou simple) et le fichier de configuration LYM_maisSens.conf généré par PG_configAndHeaderGenerator.prl
- **dans le répertoire pic** : le fichier log de l'application log-LYM-maisSens.log régénéré à chaque lancement (peut aussi contenir des captures d'écran si le header du fichier de configuration est modifié pour cela)
- **dans le répertoire scenarios** : le tableur openoffice LYM_maisSens.ods du scénario de l'application (une séquence de set-ups) ainsi que son export en format ascii (csv) LYM_maisSens.csv qui est utilisé pour la génération du fichier de configuration par PG_configAndHeaderGenerator.prl
- **dans le répertoire shaders** : les 6 shaders de l'application
- **dans le répertoire textures** : les textures de l'application – dans le répertoire

usascii la police de caractère des messages, les 7 textures d'accélération de particules, le fichier des brosses **pen.png**

Scénarios de l'application graphique

LYM_maisSens.ods

dans

C:\Users\christian\Desktop\MaisonSensibleVisuels\maisSens\maisSensPorphyrograph\scenarios

Le tableur des scénarios est composé de **colonnes** correspondant à des **paramètres** transmis à Porphyrograph (les dernières valeurs en mauve sont celles qui correspondent aux paramètres de calibration) et de lignes correspondant

- à des informations sur le type, les variables, les callbacks, les caractères d'incrément/décément de ces variables associés à ces paramètres (ainsi que leurs valeurs min et max pour ces contrôles)
- aux valeurs initiales de ces paramètres
- à des scénarios avec valeur initiale, valeur finale, et mode d'interpolation (stepwise, linear, cosine – les valeurs majuscules correspondant à des valeurs pour lesquelles sont définis un délai de lancement et une durée de variation en %age de 1)

Les premières lignes de chaque scénario correspondent aux valeurs de départ, et elles sont transférées dans Max/MSP comme valeurs envoyées vers Porphyrograph comme valeur de départ du scénario. Les variations de paramètres au cours du scénario sont implémentées comme des interpolations dans Max/MSP.

Informations techniques

Focale des vidéo-projecteurs : 0.46 correspond à un angle de vue de 90° (217cm de largeur à une distance de 100cm).

Réalisation de la scène 3D

Cube 3D mis aux dimensions de la pièce

Coordonnées de textures par *unwrap cubique*

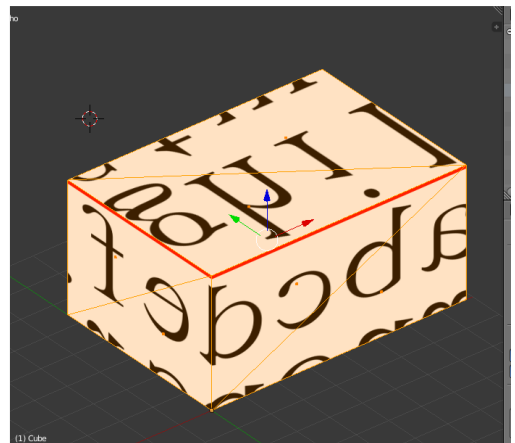
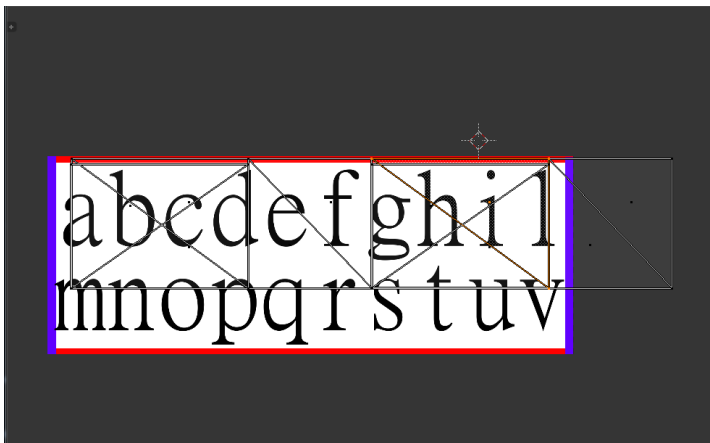
Utiliser plutôt le sticky selection mode *disabled* pour les vertex afin de pouvoir déplacer séparément les faces sans garder les connexions aux vertex.

Mettre les coordonnées de texture proportionnelles aux longueurs et placer les faces côte à côte sur la texture. Placer le plafond et le sol au dessus des murs, si possible en continuité miroir avec une des faces et une arête commune.

Raffiner la géométrie et la développer sur la texture en gardant les vertex du cube initial fixes.

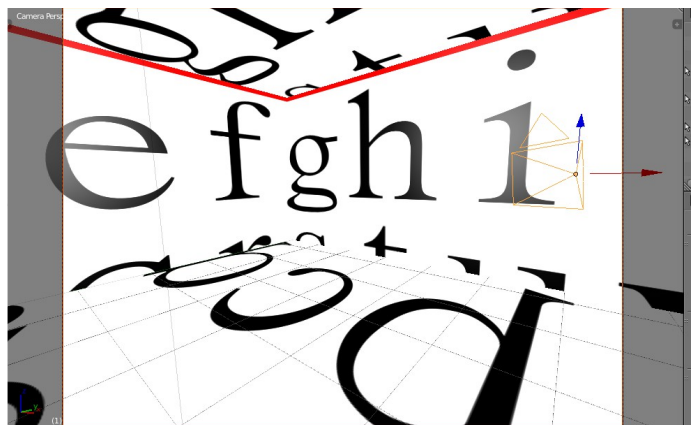
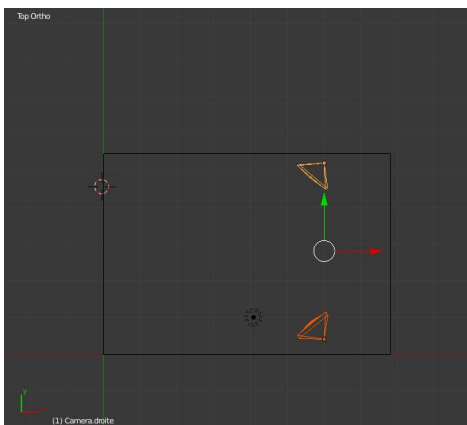
Exporter en obj avec les uv coordinates et avec option **Z up & Y forward**.

Texturage des faces avec le sol et le plafond en superposition des murs latéraux « abcd » et « ghil » (sol et plafond sont un peu moins hauts que les murs) : **à gauche** uv mapping, **à droite** résultat sur le cube texturé (la face « ef » est la face du fond où les deux VP se recouvrent).



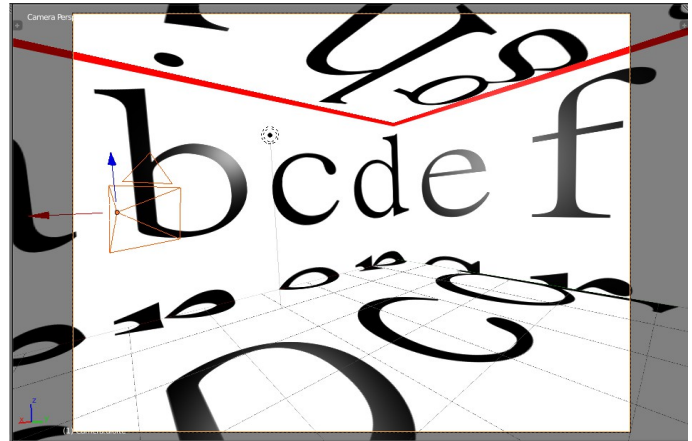
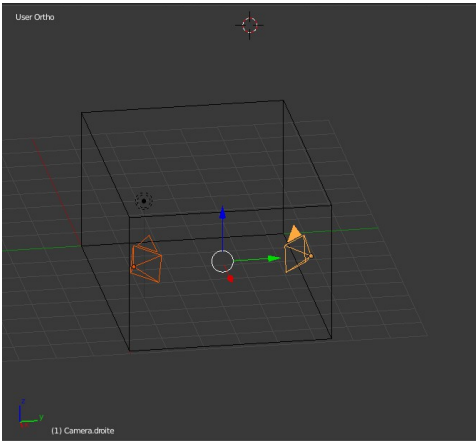
A gauche : vue de dessus avec les deux caméras (caméra de droite en jaune et caméra de gauche en orange)

A droite : vue de la caméra de gauche



A gauche : vue de dessus avec les deux caméras (caméra de droite en jaune et caméra de gauche en orange)

A droite : vue de la caméra de droite



Paramètres de calibrage

Positionnement de la caméra

$(projLocX, projLocY, projLocZ)$: position de la caméra en 3D

$(projCenterX, projCenterY, projCenterZ)$: point central vers lequel regarde la caméra

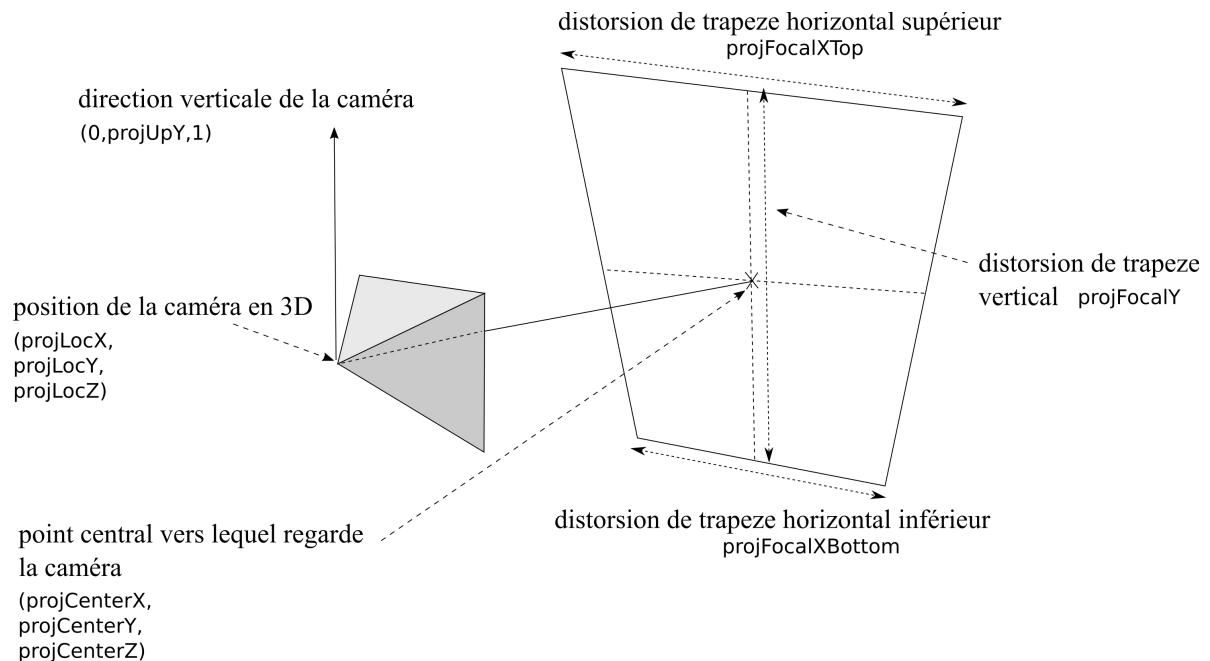
$(0, projUpY, 1)$: direction verticale de la caméra

Paramètres intrinsèques de la caméra

$(projFocalXTop, projFocalXBottom, projFocalY)$: distorsion de trapeze, ces valeurs permettent d'ajuster l'image en introduisant des distorsion de trapèze additionnelles afin de mieux prendre en compte les paramètres intrinsèques du projecteur.

$projWidthAt1m$: la largeur de projection observée à 1 m afin de définir la focale du projecteur

$projRatio$: le ratio largeur sur hauteur

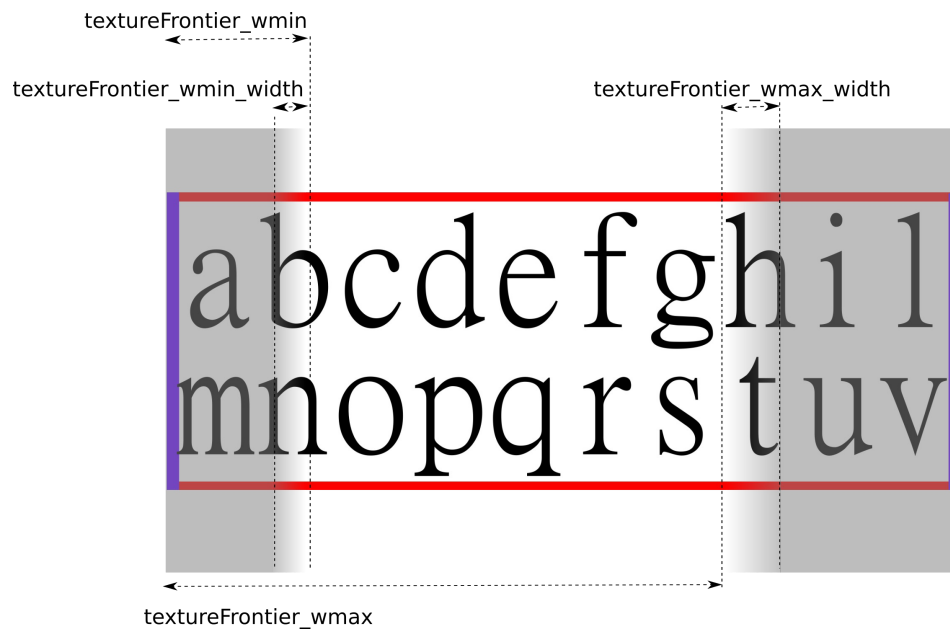


Paramètres de gestion des bords de texture

$textureFrontier_wmin$: la coordonnée de texture horizontale a laquelle les particules commencent à être visibles

$textureFrontier_wmax$: la coordonnée de texture horizontale a partir de laquelle les particules commencent à ne plus être visibles

`textureFrontier_wmin_width` : la largeur de la bande de texture d'interpolation entre texture visible et noir (pour la coordonnée `textureFrontier_wmin`)
`textureFrontier_wmax_width` : la largeur de la bande de texture d'interpolation entre texture visible et noir (pour la coordonnée `textureFrontier_wmax`)



`meshFrontier_Y` : la coordonnée en y qui définit la limite de recouvrement entre le rendu des deux maillages
`meshFrontier_width` : la moitié de la largeur de la zone de recouvrement entre les deux maillages

