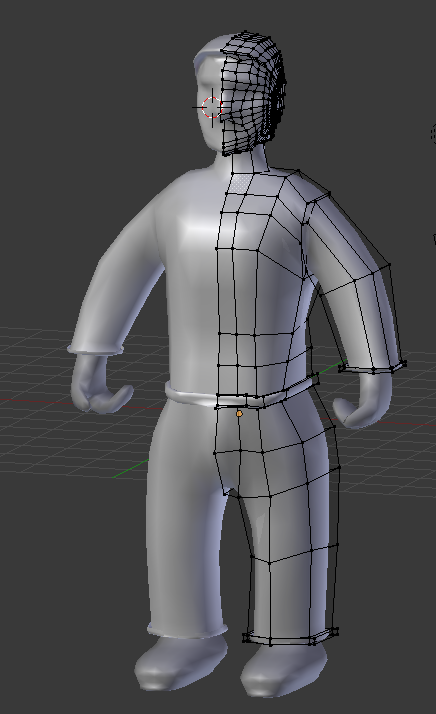
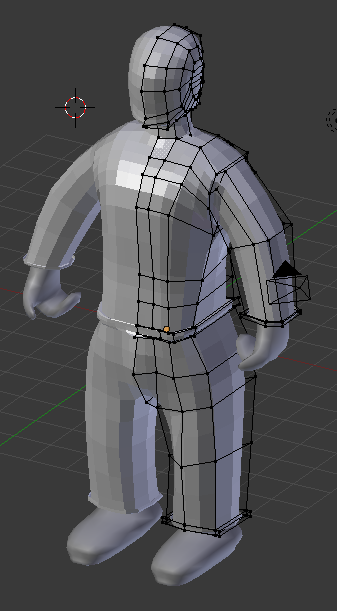
Modélisation et Animation

# Modélisation du personnage

Nous avons créé un personnage, celui-ci n’as pas était garder par la suite car l’implémentation du personnage du ninja était déjà terminer et plus complète que notre modèle. De plus, il fallait mettre en place l’animation du chat.

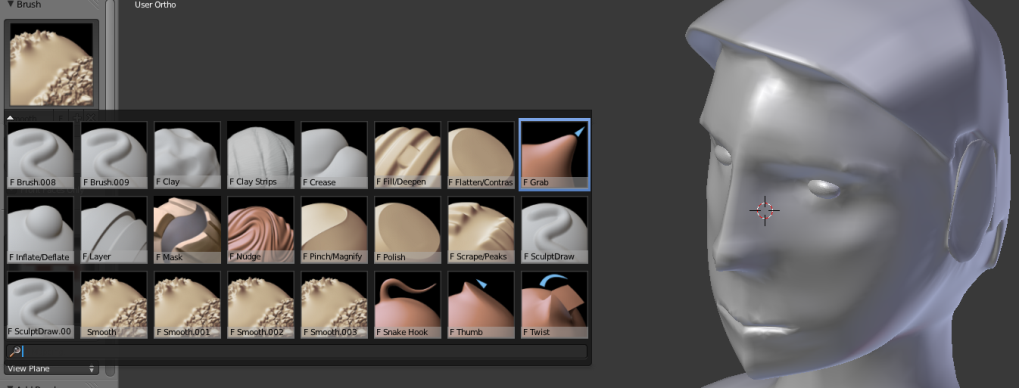
Cependant, ce travail m’as permis de voir comment créer un personnage. Pour cela, on formera dans un premier temps la structure de notre personnage de manière grossière :



Puis on ajoutera au fur et à mesure des séparations afin d’augmenter le niveau de détails (notamment pour la tête du personnage). On fera le tout en appliquant des modificateurs dans Blender, celui de la symétrie pour avoir à faire seulement une partie du corps et le modificateur « sub-division surface » afin que notre modèle soit plus lisse en ajoutant un nombre de surface supplémentaire(tout en gardant un niveau bas pour éviter d’avoir trop de face). Le corps et la tête du personnage sont en un seul morceau et sont fabriquer grâce à plusieurs extrusions. Seul les bras et les pieds sont séparés, on les ajoutera par la suite.

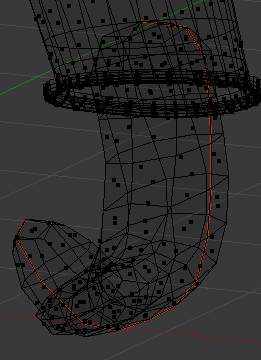
Cette partie de la modélisation est très longue, il faut ajouter les parties petit à petit tout en essayant de garder un aspect au personnage.

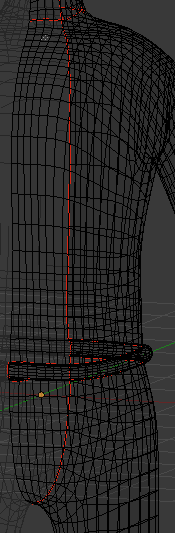
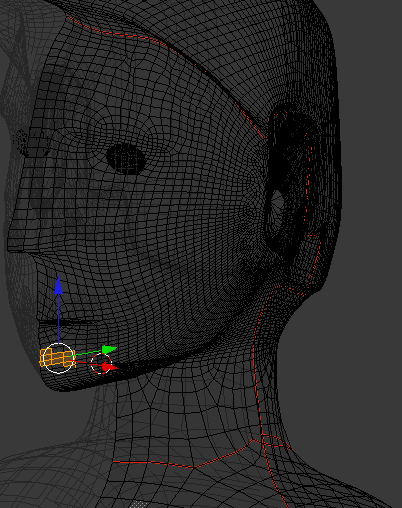
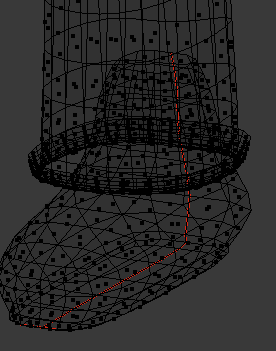
Une fois que notre personnage possède des formes de base, on appliquera le modificateur de sub-division afin d’obtenir les vertex réel de notre personnage. On pourra alors, passer en mode sculpture, ce mode permet de pouvoir sculpter notre objet avec différents outils pour lui donner la forme que l’on souhaite, cette outil est intéressant mais nécessite beaucoup de face pour faire du travail précis ce qui n’est pas bénéfique pour le rendue dans ogre. Cependant, on pourra s’en servir afin de marquer le nez, la bouche, les yeux.



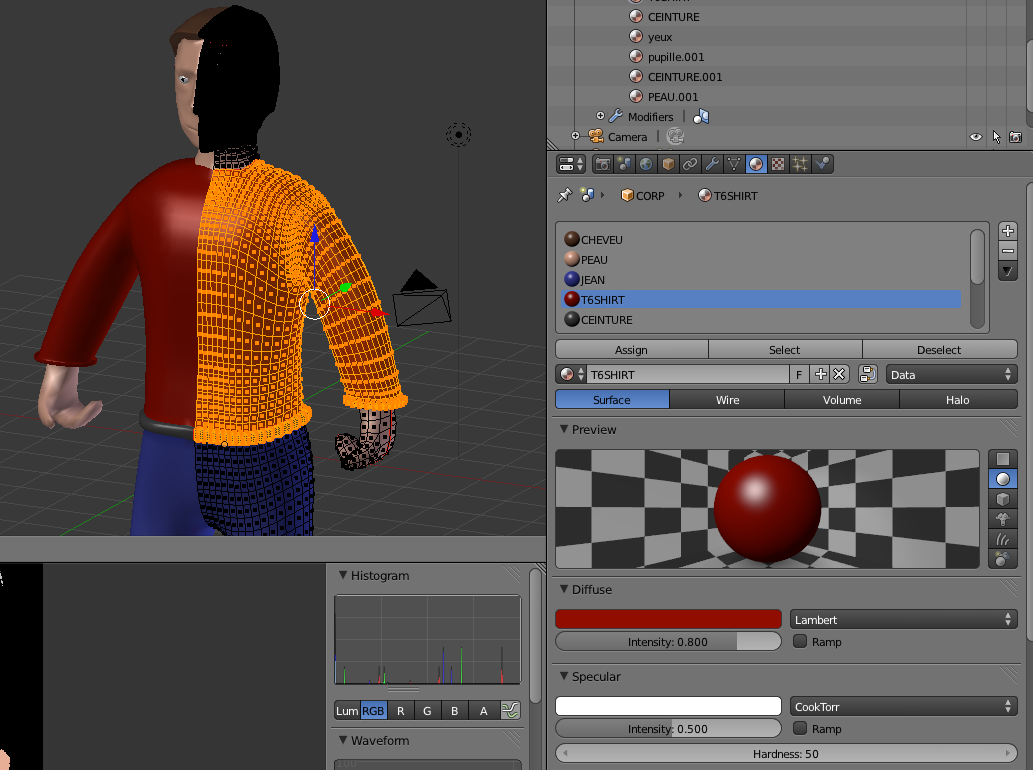
Pour la création des yeux la solution la plus simple est de créer une sphère à laquelle on peut enlever la moitié. Pour mettre en place les textures, il va falloir maintenant ajouter nos mains, les pieds et les yeux en joignant le tout avec ctrl+j. A ce stade, nous avons plus qu’un seul mesh pour notre personnage.

## Texture du personnage

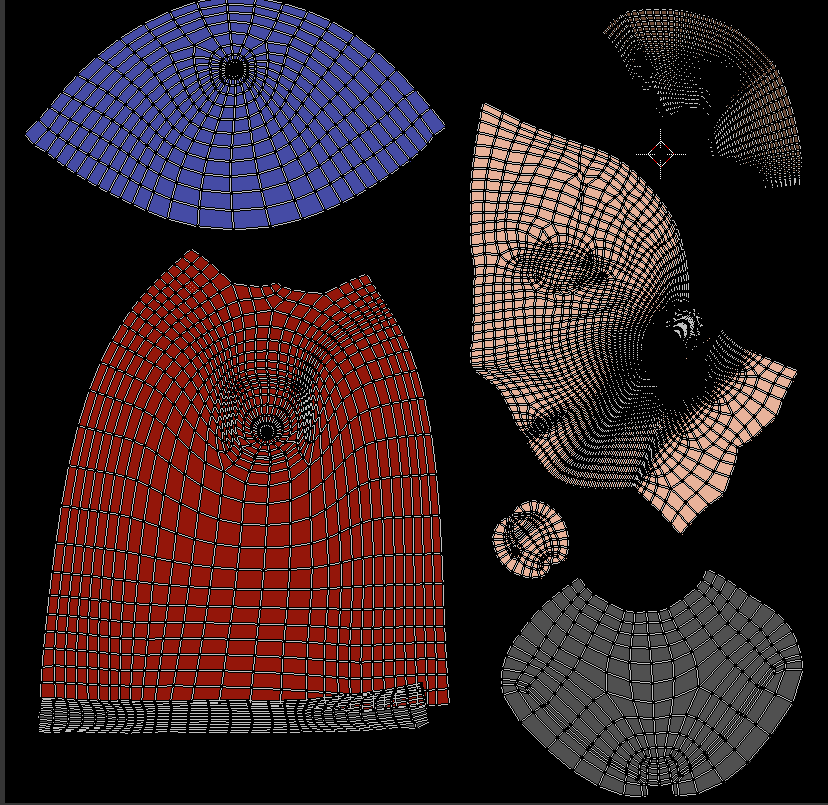
Pour la mise en place de la texture, on va définir des zones de découpages, celle-ci sert lors du dépliage de notre texture sur ce que l’on appel l’UV map. Pour cela, on ce met en edit mode et on choisira les vertex de séparation. Dans notre cas ces vertex ce situe au niveau du cou (séparation tête et T-shirt), la chevelure, la ceinture et le long du dos, du cou, des mains et des pieds pour séparé la texture.

Une fois les séparations effectuées, on peut lui appliquer des matériaux. Cela simplifiera la mise en place de la texture en attribuant une première couleur aux différentes parties. Pour cela, on ce met en edit mode et avec le mode de sélection des faces puis on va appuyer sur la touche L afin de sélectionner les différentes parties prédécoupé. Enfin, en allant dans le menu des matériaux on ajoute le matériau ou on en choisie un existant à appliquer à notre model.



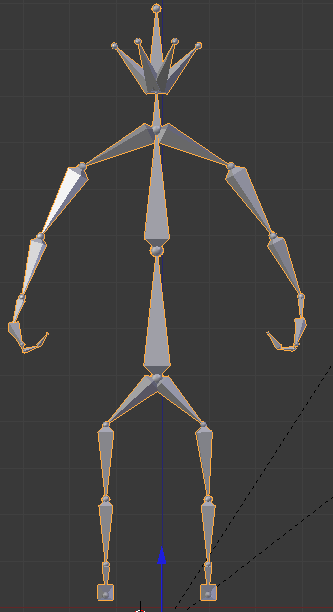
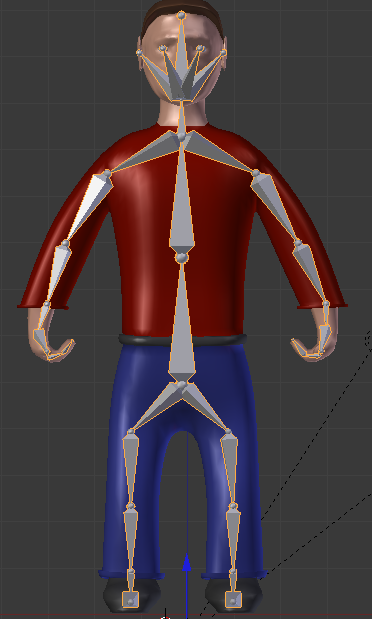
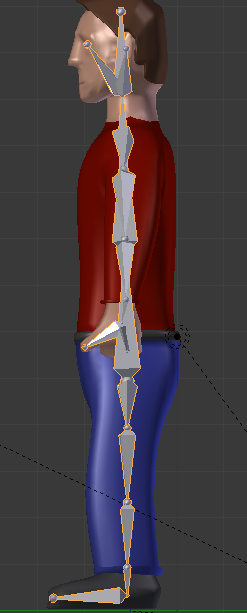
Ensuite, il faudra déplier et appliquer nos matériaux sur la texture. Pour cela, on appuie sur « U » et on choisit « unwrap ». Cela va afficher dans l’UV editor nos formes dépliées, il faudra les replacer correctement. Par la suite on applique les matériaux définies auparavant en cliquant sur « bake » avec l’option texture afin d’appliquer les couleurs des matériaux sur la texture :

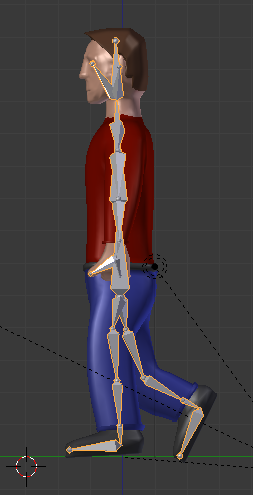
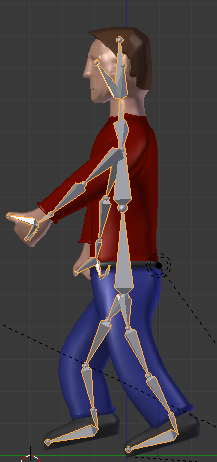
On pourra alors par la suite redessiner sur la texture comme on le désire.

## Animations du personnage

Pour mettre en place les animations, il faut définir un squelette. Pour cela, on va placer dans notre modèle un os en bas du dos et au travers de différentes extrusion agrandir notre squelette sur chaque extrémité. On obtient alors un squelette qui correspond aux formes de notre personnage :

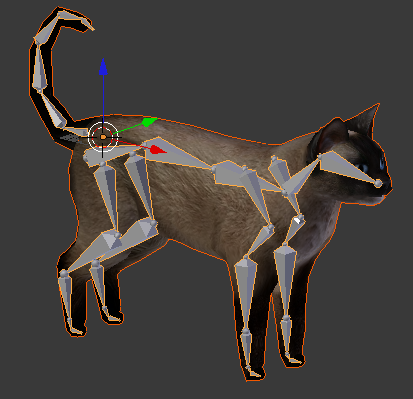
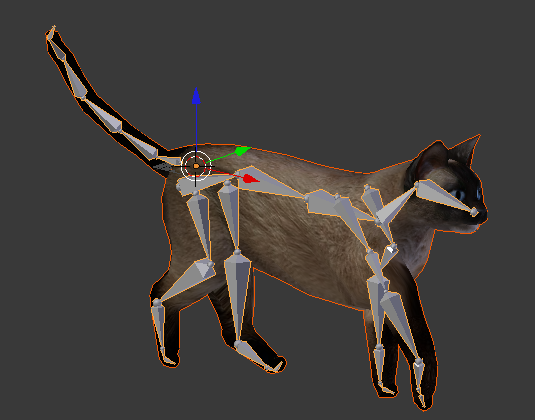
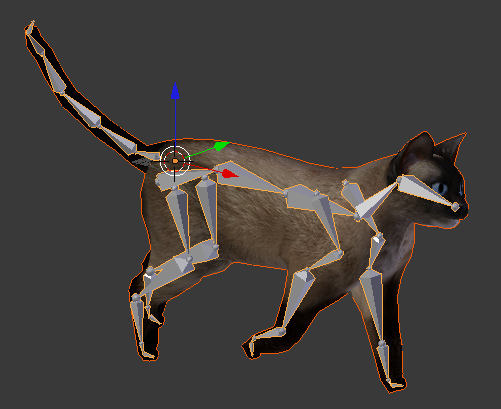
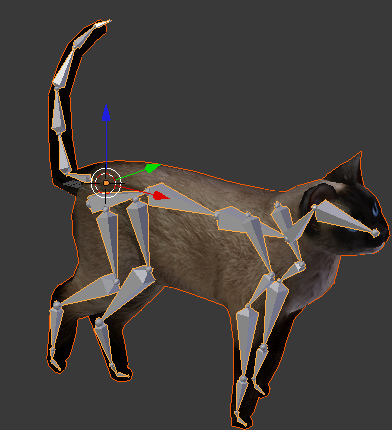
Une fois notre squelette correctement positionné on peut après avoir fait ctrl+p et choisir le mode de poids automatique passé en mode « pose ». Dans ce mode on peut faire bouger les articulations de notre modèle et grâce à la touche « i » marquer la positionner actuel comme position clé (KeyFrame). On peut alors définir notre animations en déplacent les os sur plusieurs frames.

## Animation du chat

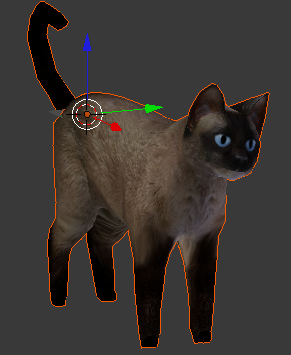
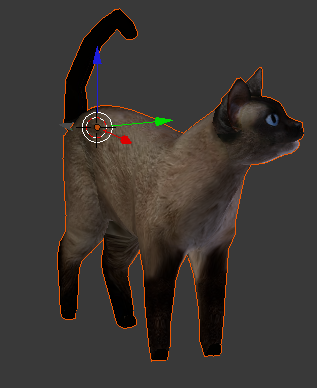
Nous possédions un model de chat, celui-ci avait déjà le squelette et les textures de définie. Il ne rester plus qu’a mettre en place des animations. De la même manière que pour le personnage on en pose mode définir nos keyframes afin de définir notre animation. J’ai mis en place 4 animations : attendre, marcher, assoir, debout. Dans le programme seulement 2 seront mis en place.

Pour l’animation de marche, il y a une astuce, un chat quand il se déplace bouge de manière coordonnée la pate avant gauche avec l’arrière droite et inversement. Du coup, on définira des frames avec les extrêmes et les positions intermédiaires (pate aligné) c'est-à-dire la pate le plus avancée possible – position intermédiaire – pate le plus reculé possible – position intermédiaire :

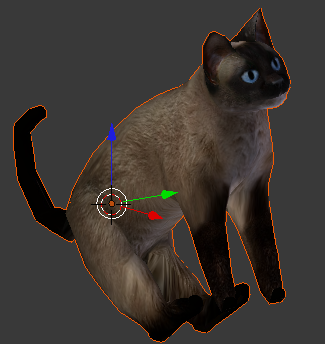
     

Une fois les Keyframe mis en place, on aura plus qu’a ajusté la longueur de l’animation en déplaçant comme désiré les frames. De plus, pour une animation plus réaliste il est intéressant de multiplier la longueur de l’animation en rajoutant un cycle de marche supplémentaire. Enfin, pour rendre le tout encore plus réaliste on animera la tête et la queue du chat.

L’attente du chat consiste juste à bouger la tête et la queue, il suffit donc de quelque frame avec la tête et la queue dans différentes position.

De même pour s’assoir et se lever, l’animation est la même dans le sens inverse, il suffit alors de faire assoir le chat sur ces pates arrières :



Enfin, il faut exporter le tout en format ogre en vérifiant (très important !) que le mesh et le squelette de notre animation ont bien le même point d’origine.

## Implémentation du chat dans ogre

Pour implémenter notre chat dans ogre, on l’ajoutera dans la scène. On créer donc une fonction setChat() qui placera le chat dans un nœud du scène de graphe, on vérifiera les proportions de notre chat et sont emplacement.

On ira ajouter dans la classe des animations, les animations correspondant au 4 animations définie auparavant avec les fonctions suivantes : walkCat(), waitCat (), sitCat , getupCat().

Chacun de ces fonctions appellant l’animation correspondante exemple :

Entity\* chat = mShortStory->getSceneManager()->getEntity("Chat");

chat->getAnimationState("marche")->setEnabled(true);

chat->getAnimationState("marche")->addTime(evt.timeSinceLastFrame);

Enfin il faut pouvoir faire en sorte que le chat suive le personnage du regard et qu’il le suive. Pour cela, dans la fonction d’animation InputListener::frameRenderingQueued on appliquera les différents mouvement ainsi que l’appel au fonctions déterminant l’animation à exécuter.

Pour le déplacement, il suffit de calculer le vecteur directeur dans le plan X,Z pour cela on effectue un simple calcul vecteur\_déplacement = personnage\_position – chat\_position puis on normalise ce vecteur il suffit alors de translater le mesh dans cette direction en multiplier par un facteur de vitesse .De plus, on vérifiera que le mesh garde une certaine distance pour ne pas coller le personnage. Pour ajouter les animations il suffiera d’appeler notre fonction d’animation correspondant à quand le chat marche ou attend selon la condition de distance.

if((fabs(pers\_pos.x - chat\_pos.x) > 200)||(fabs(pers\_pos.z- chat\_pos.z) > 200)){

mScene->getCatNode()->translate(move\_cat\*4);

mAnimation->walkCat(evt) ;

}else{

mAnimation->waitCat(evt);

}

Pour finir, il faut que le chat regarde vers le personnage car il se déplace vers lui. Pour cela il faut récupérer l’orientation du mesh selon l’axe X puis de définir le quaternion correspondant à l’angle de rotation pour que le chat reste en face du personnage et enfin appliquer cette rotation à notre mesh.

Ogre::Vector3 orientation = mScene->getCatNode()->getOrientation() \* Ogre::Vector3::UNIT\_X;

Ogre::Quaternion quat = orientation.getRotationTo(move\_cat);

mScene->getCatNode()->rotate(quat);

Pour conclure, la modélisation et l’animation prennent beaucoup de temps, notamment lorsque les modèles sont complexe. De plus, ce personnage étant mon premier il a fallut du temps pour obtenir une forme à peu près réel et surtout la mise en place de la texture qui est loin d’être facile.