Sandra Lavoie

Étape 1 : Créer le cube prototype

Il est beaucoup plus simple de créer un cube qui fonctionne à merveille, sur lequel on base toute la programmation pour ensuite le dupliquer en quantité voulu.

Il faut donc créer un premier mesh et le positionner en location et rotation X,Y et Z = 0 (c'est seulement plus simple par la suit pour dupliquer)

ADD – MESH – CUBE

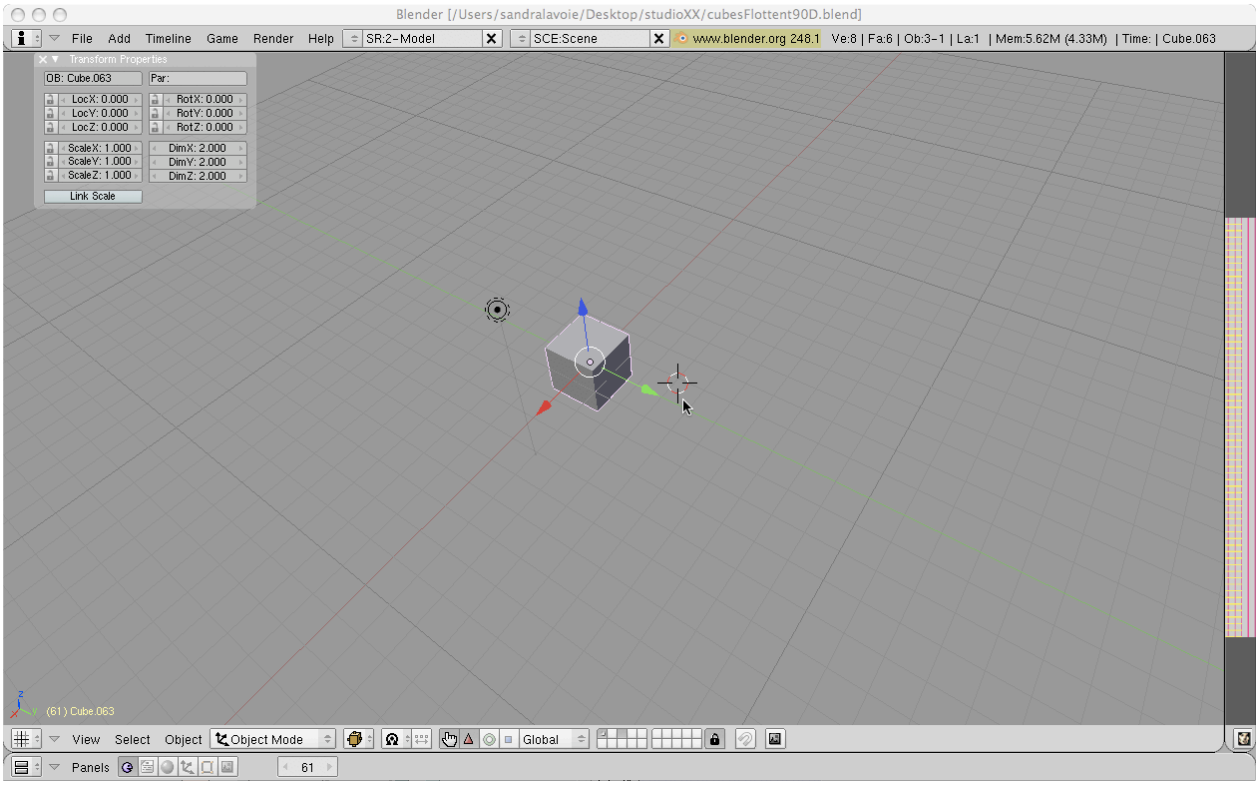

Figure A

Étape 2 : Créer un effet de flottement

Afin de simuler un espace où vivent les cubes, nous allons créer un effet de flottement dans le vide de l'univers.

Créer une deuxième fenêtre à la droite du *3D view* et dans le menu *Window type* sélectionner le *Text Editor*

Il faut débuter chaque scipt en y important le game logic de Blender (lignes 1 à 3 en Figure B)

Pour créer le flottement nous allons créer deux propriété : «gateh» et «gateb»

Sandra Lavoie

```
import GameLogic as g
c = g.getCurrentController()
obj = c.getOwner()

obj.test = 2
#objPosi = obj.getPosition()
(x,y,z) = obj.getPosition()

#obj.y = z
# Pour créer le flottement, il faut simuler un random sur les
# propriétés gateh et gateb

obj.gateh = 0
obj.gateb = 0

# L'effet de flottement peux seulement avoir lieu si le cube
# se situe au dessus de la heuteur de 0.4

if z > 0.4:
    obj.gateh = 1
    obj.gateb = 0

if z < 0.4:
    obj.gateb = 1
    obj.gateh = 0
```

Figure B

Dans le game logic nous allons programmer un random qui donnera des petites pousser sur le cube vers le haut et vers le bas pour simuler le flottement. (voir figure C) À noter que toutes les transformation dans la fenêtre *Motion* ne doivent pas être en *Local Transformation* (le petit L à coté de la location ne doit pas être enclenché).

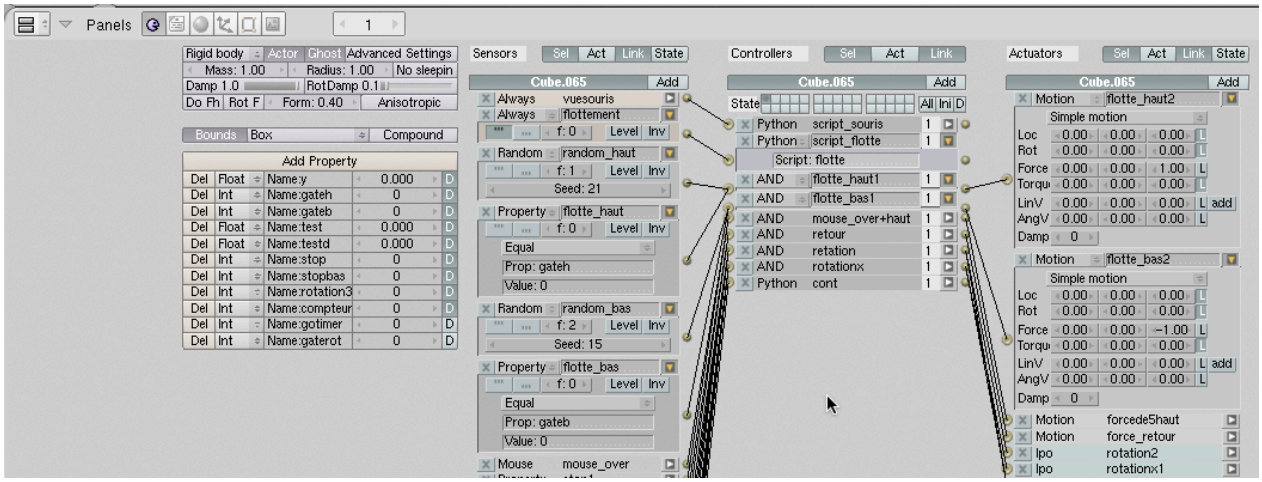

Figure C

Vérifier dans la simulation de jeu (Dans la fenêtre *3D view* appuyer sur P sur le clavier pour activer le jeu et ESC pour arrêter) que le cube se déplace seul légèrement de bas en haut. L'effet seras plus réussit lorsque plusieurs cubes flotterons côte à côte.

Étape 3 : Afficher la souris

Avant de pouvoir créer de l'interactivité, il faut afficher la souris dans le jeu. Dans le Text Editor, créer un nouveau scipt (Figure D) et l'activer dans le game Logic (Figure E).

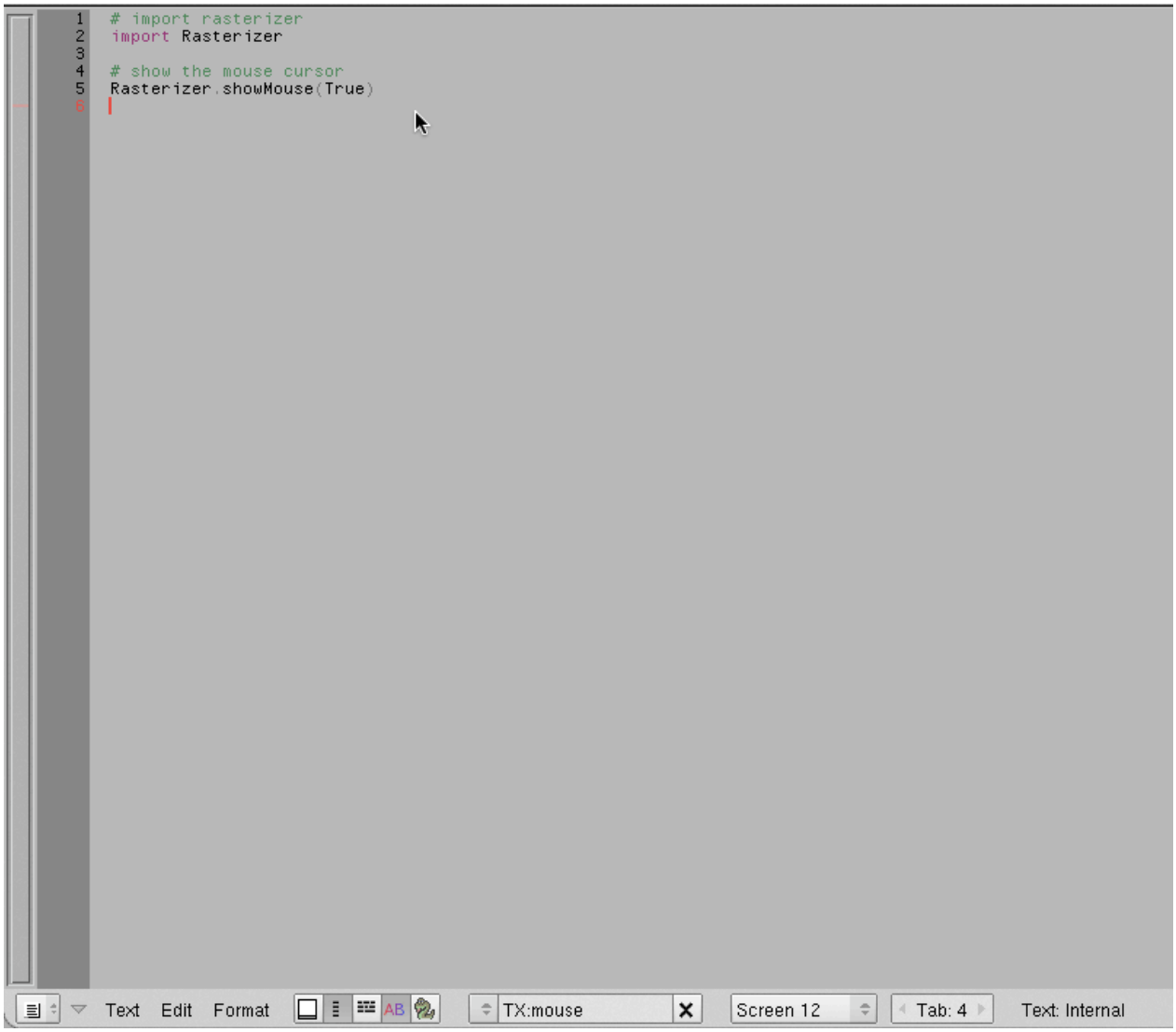

Figure D

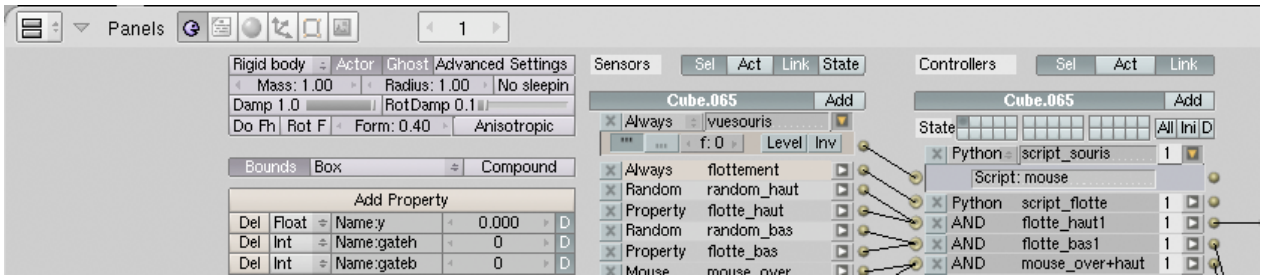

Figure E

Sandra Lavoie

Étape 4 : Faire élever le cube

Dans le GameLogic programmer que lorsque la souris est dessus le cube (en mouse over), une pousser est donner à celui-ci d'une force de 5 vers le haut (5 sur l'axe Z). Ensuite, lorsque la souris n'est plus sur le cube (mouse over Inverse), une pousser de 3 lui est donner vers le bas (-3 sur l'axe Z). (voir Figure F) Toujours vérifier que le bouton *Local Location* n'est pas enfoncé.

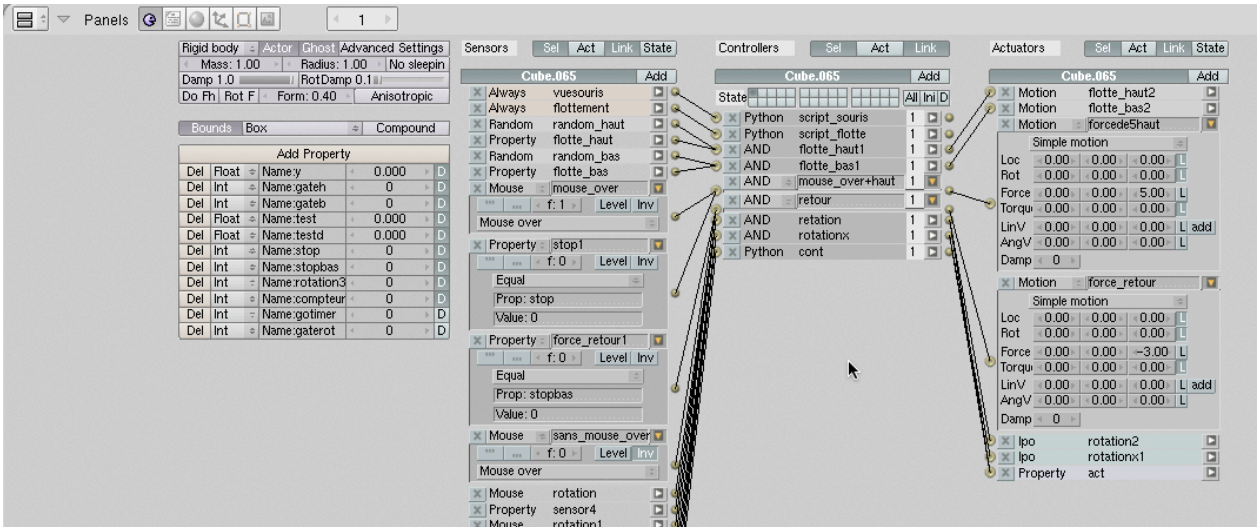

Figure F

Bien sur, il faut programmer le tout dans le script afin d'arrêter les pousser à une certaine hauteur (figure G). À noter que les figures B et G sont sur le même script. Il est mieux de réduire le nombre de script afin d'alléger le taille du document.

```
    obj.gateh = 0

# Le cube ne peut plus recevoir de pousser vers le haut s'il est à une
# hauteur plus grande que 5

if z > 5 :
    obj.stop= 1


if z < 5 :
    obj.stop= 0

# Le cube ne peut plus recevoir de pousser vers le bas s'il est à une
# hauteur plus basse que 1.5

if z > 1.5 :
    obj.stopbas= 0


if z < 1.5 :
    obj.stopbas= 1
```

Figure G

Vous pouvez vérifier que le tout fonctionne dans la simulation de jeu.

Étape 5 : Faire tourner le cube

Sandra Lavoie

Cette étape est plus complexe. Nous allons d'abord empêcher le cube de tourner avant qu'il soit à une hauteur minimum de 2 sur l'axe Z afin qu'il ne culbute pas les autres cubes. (Toujours sur le même script qu'en B et G)

```
46
47  # Le cube peut tourner seulement s'il est à une hauteur plus élevé que 2
48
49  if z > 2 :
50      obj.rotation3= 0
51
52
53  if z < 2 :
54      obj.rotation3= 1
```

Text Edit Format TX:flotte Screen 12 Tab: 4 Text: Internal

Figure H

Nous allons d'abord programmer les rotations sur l'axe Z et Y dans le Game Logic et ensuite animer leur mouvement en IPO curves.

Avec un clic du bouton de gauche de la souris sur le cube, il seras possible de faire tourner le cube sur lui même sur l'axe des Y. (Game Logic en Figure I et IPO en figure J)

Le IPO appelé «rotation 2» active les animations de rotation sur l'axe des Y en frames 1 à 60. Vous pouvez jouer avec la courbe afin de créer une animation plus fluide. Assurez-vous que le bouton *Add* est enfoncé dans les *actuators* dans le Game Logic. Ce bouton permet que chaque animation va s'additionner une à l'autre.

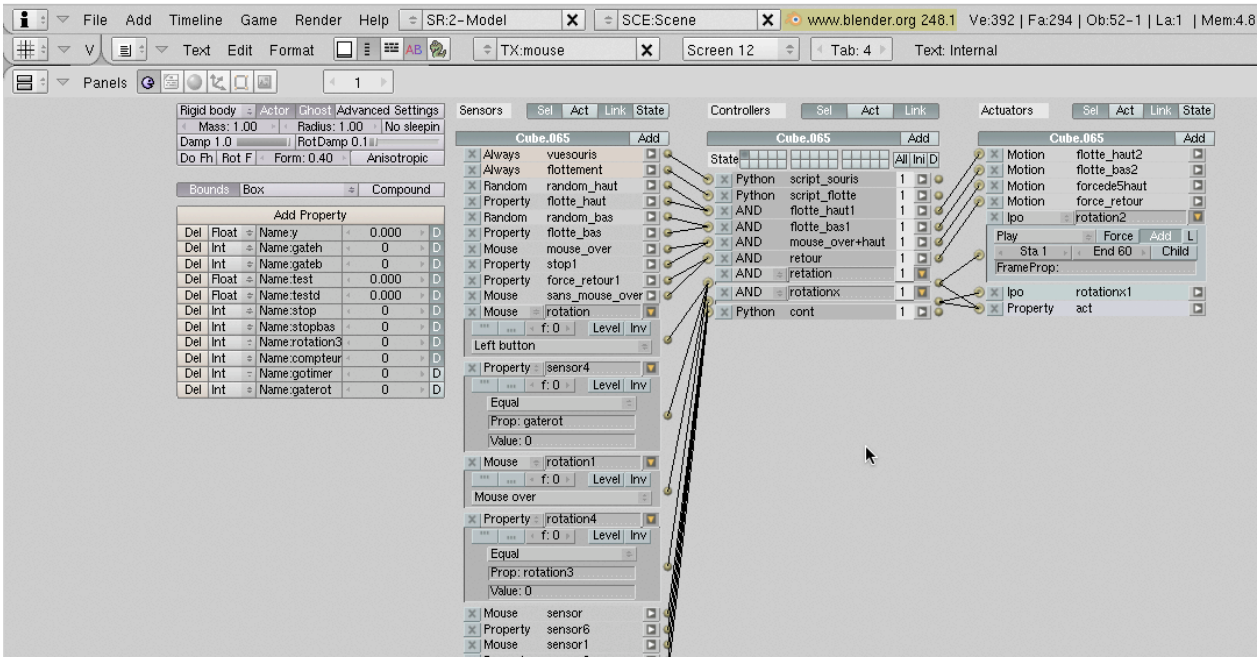

Figure I

Sandra Lavoie

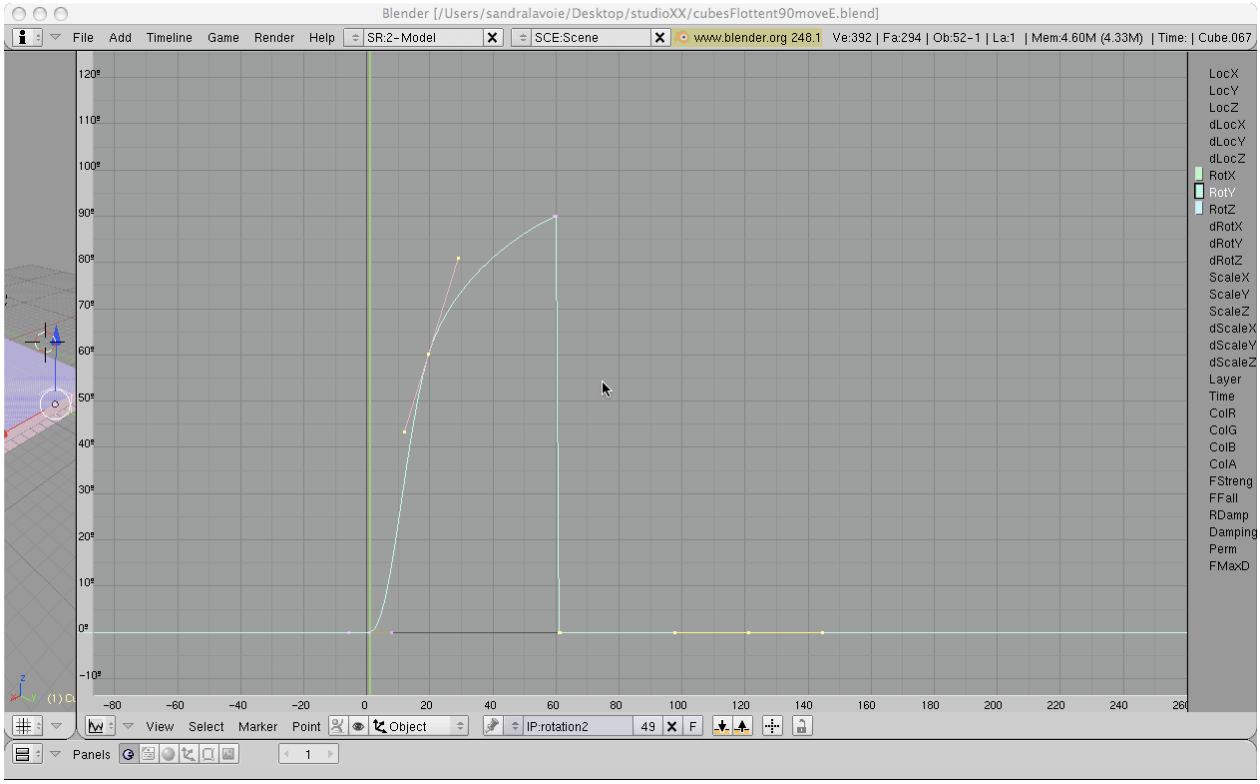

Figure J

Faire la même chose pour les rotations en X avec le clic du bouton droit de la souris sur le cube. Le IPO appelé «rotation 2» active les animations de rotation sur l'axe des X en frames 61 à 120.

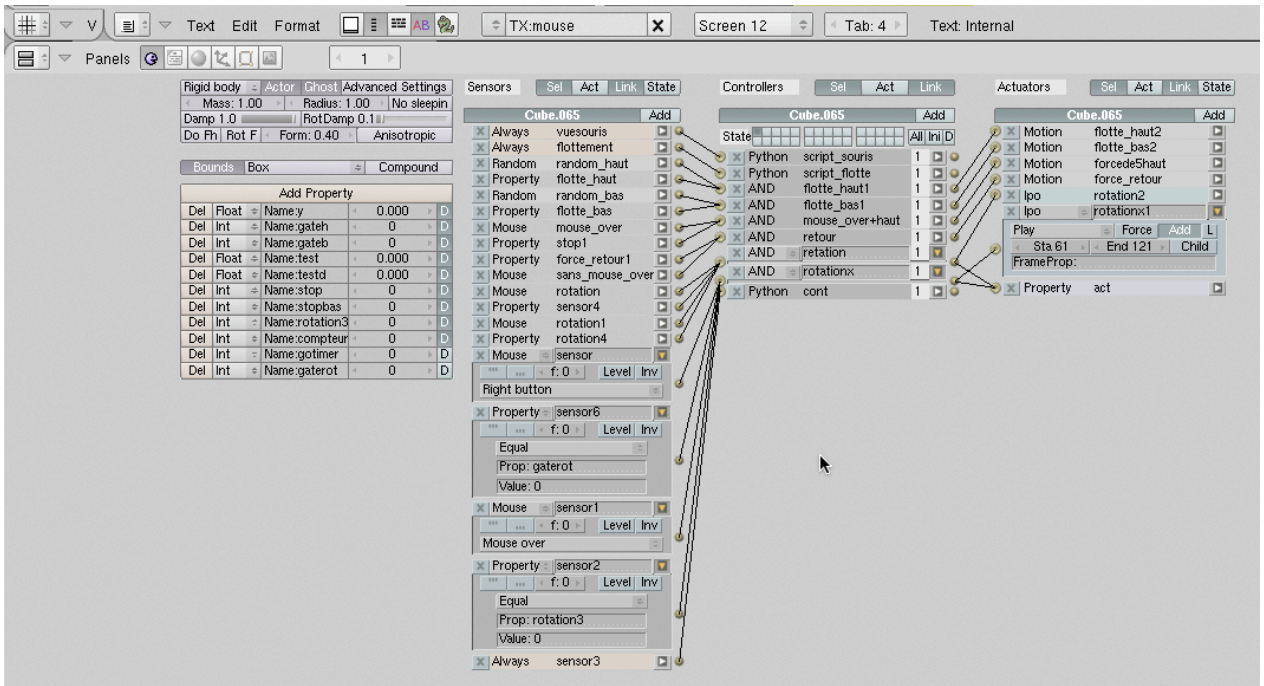

Figure K

Sandra Lavoie

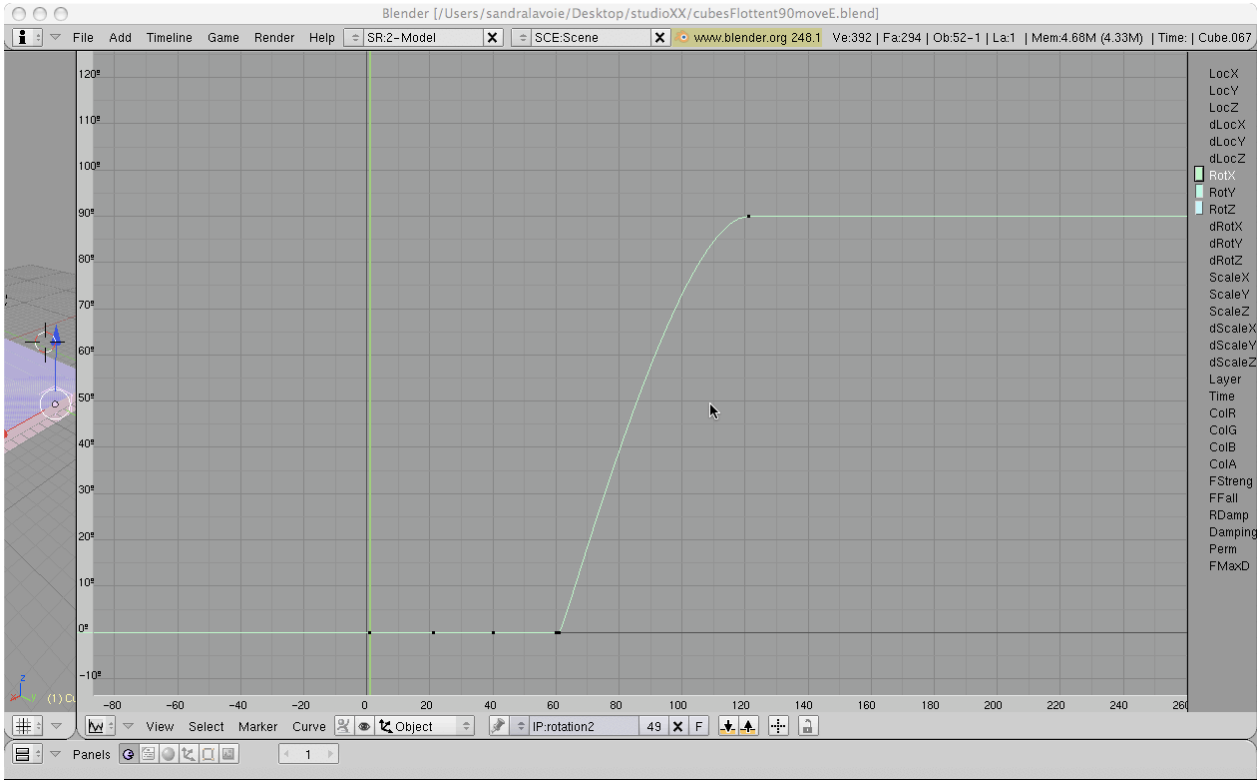

Figure L

Afin d'empêcher les cubes de s'entrechoquer et de faire des rotations non complètes, il va falloir créer un *timer* qui va limiter le moment pour activer les animations. (Ici, le temps du *timer* à été défini par essais et erreurs jusqu'au résultat voulu.)

Portez une attention toute particulière aux connections dans le Game Logic puisque parfois plus d'un *Sensors* ou *Actuators* est connecté aux *Controllers*. Cela peut sembler complexe mais il s'agit de comprendre la logique.

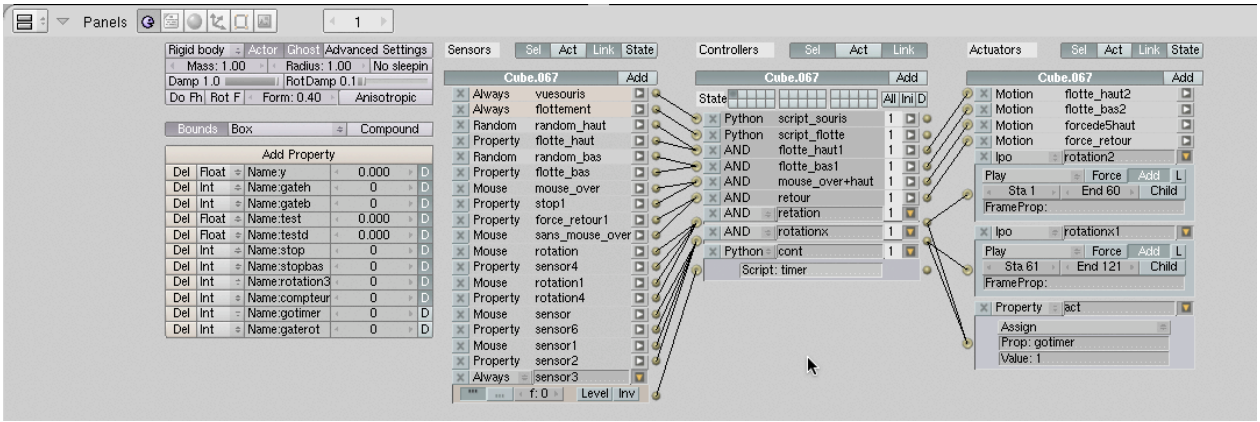

Figure M

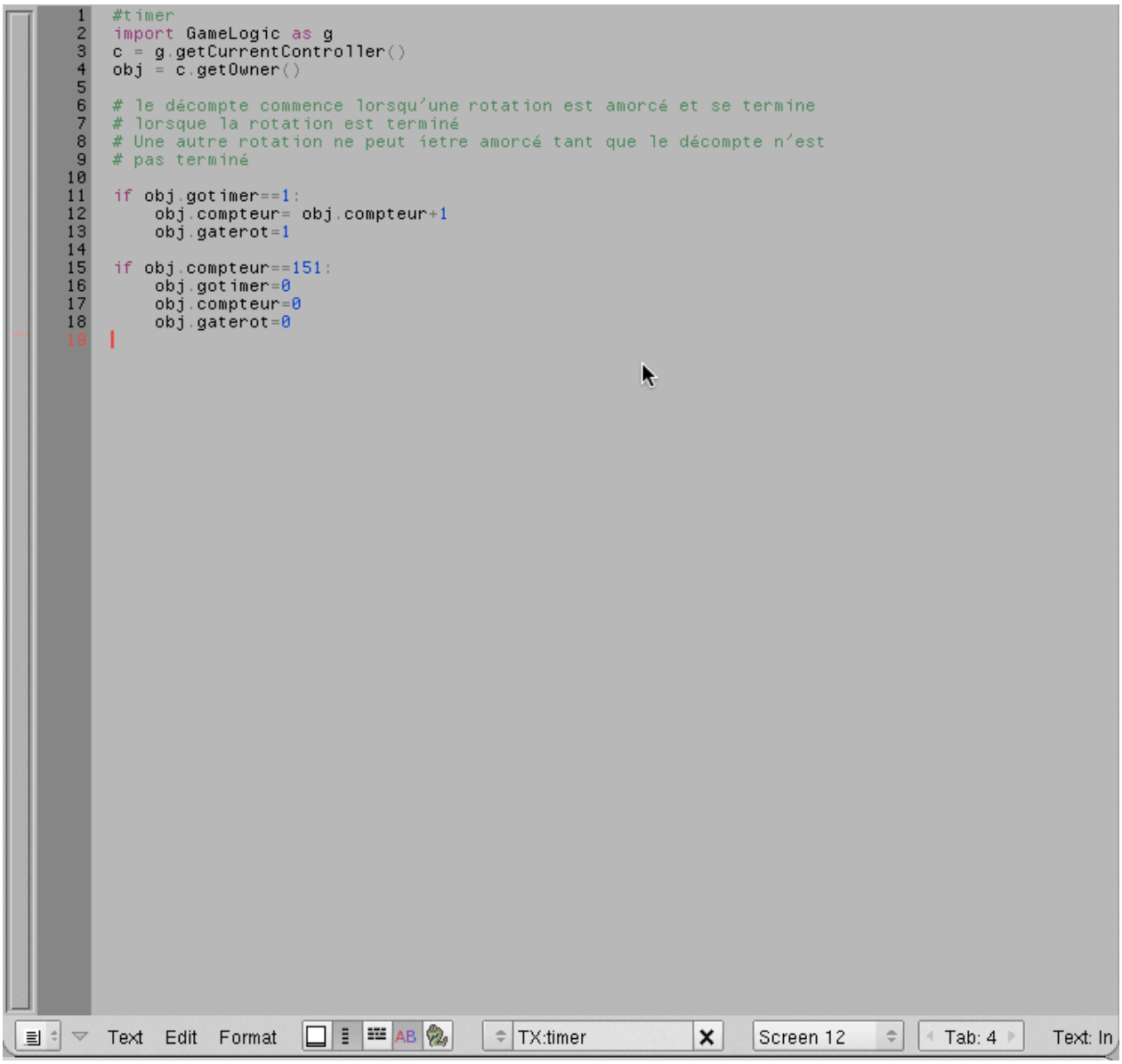

Figure N

Étape 6 : Mapping

Dans ce projet, tous les cubes ont les mêmes surfaces, il est donc plus rapide de faire le mapping avant de dupliquer le cube. Pour un projet où chaque cube aurait ses images de surface uniques, il faudrait par contre faire le mapping cube par cube. Mais une bonne préparation des surfaces dans le *UV/images Editor* facile grandement le travail.

Dans un logiciel pour travailler les images, se préparer un document de 1024 X 1024 pixels. Ici, j'ai décidé que chaque face du cube aura une taille de 128 x 128 pixels. J'ai apposé l'image de chaque face une à côté de l'autre sur mon document de 1024 pixels^2 qui deviendra mon *mapping*. (figure O)

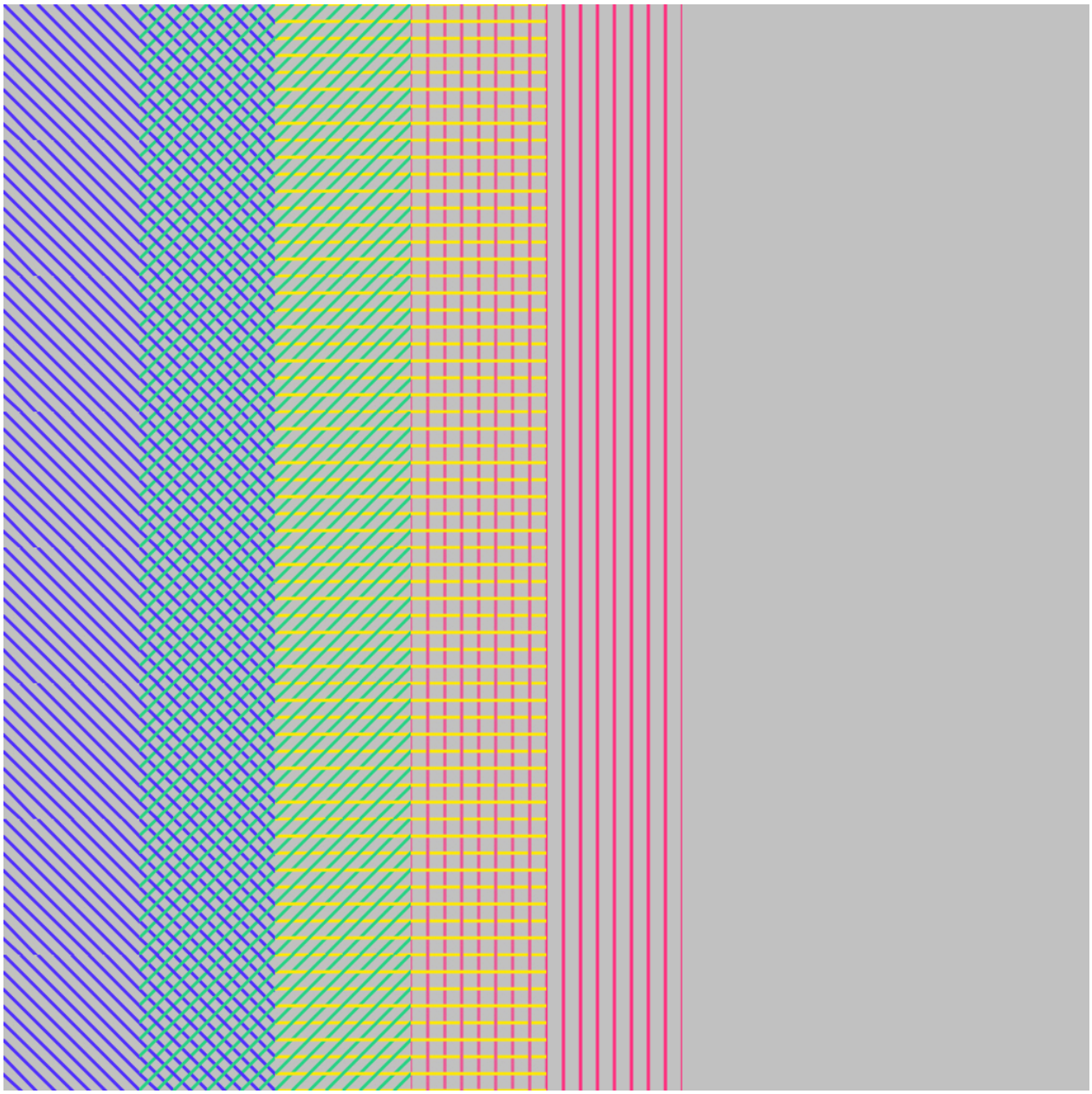

Figure O

Ouvrir le document dans la fenêtre *UV/images Editor* (image – Open) et bien délimité les surfaces de notre cube en sélectionnant celui-ci en *Edit Mode* dans la fenêtre *3D view* et ensuite retourner dans la fenêtre *UV/images Editor*. Vous pouvez donc *node* par *node* placer la surface sur la *map*.

Sandra Lavoie

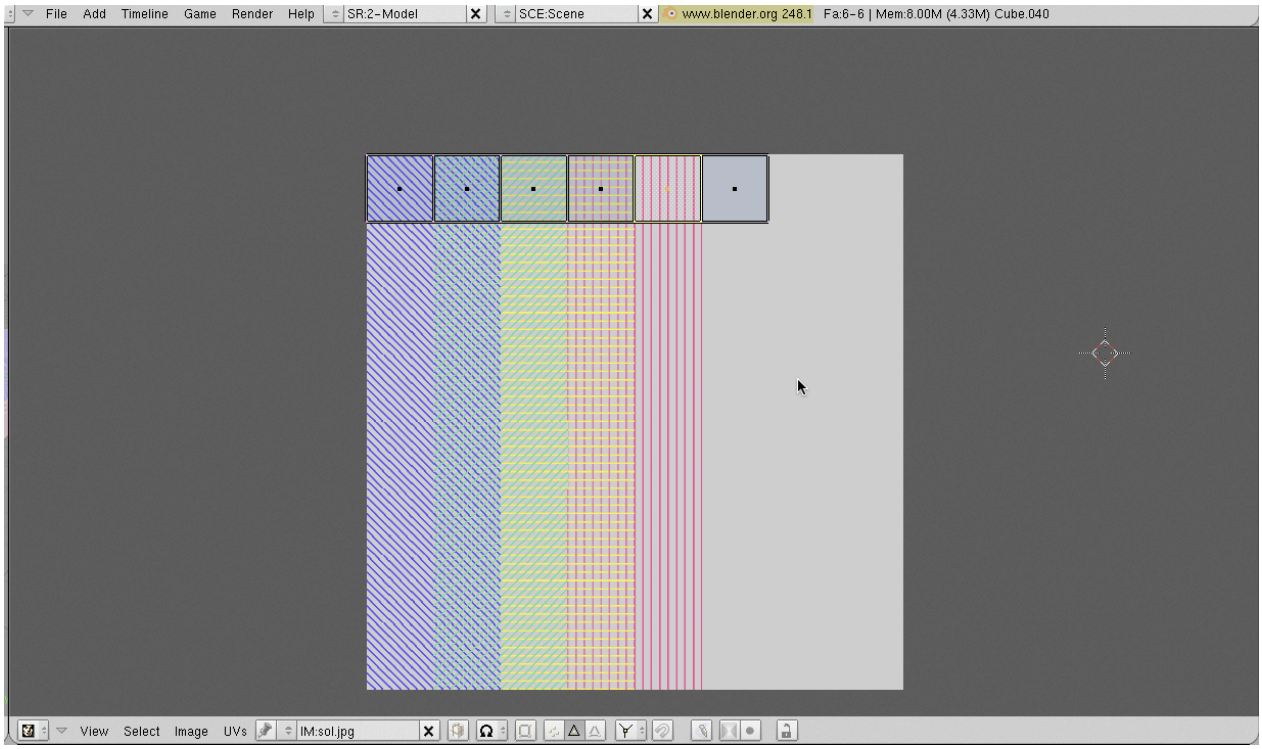

Figure P

Afin d'appliquer la *map* sur le cube s'assurer que le bouton *texFace* est enfoncé dans le *matériel* dans la fenêtre *Shading*. (figure Q)

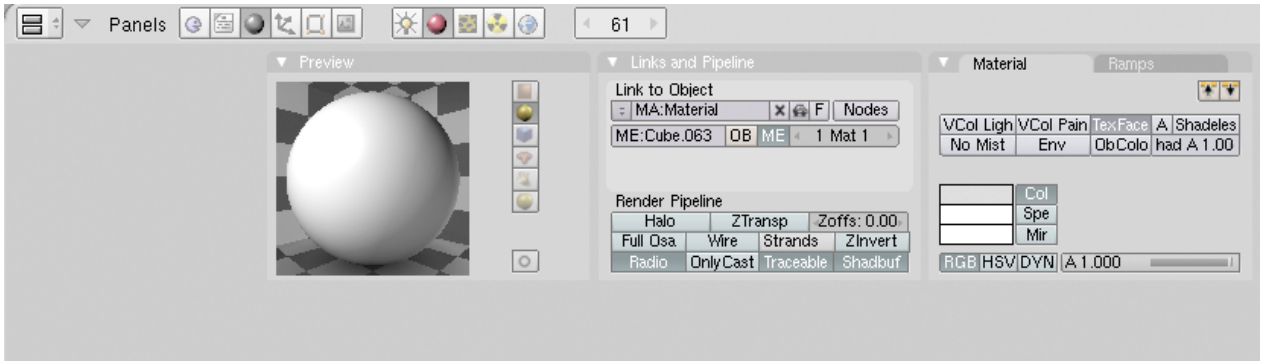

Figure Q

ÉTAPE 7 : DUPLIQUER

ÉTAPE 8 : CAMÉRA

ÉTAPE 9 : RENDU FINAL