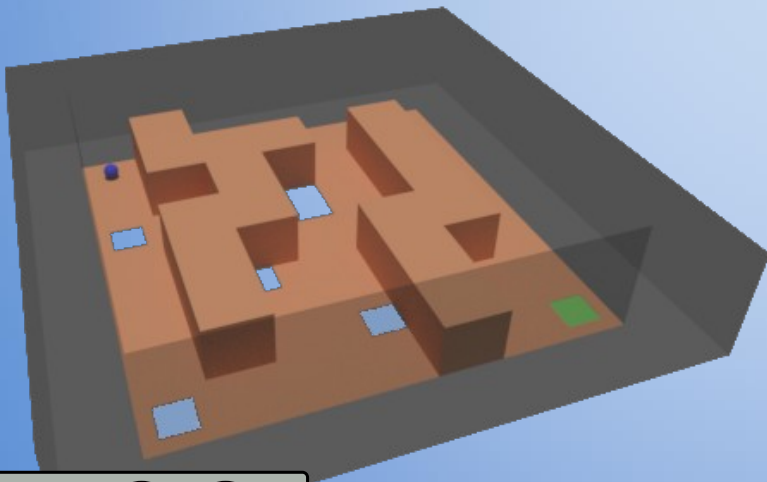


# Labyrinthe à bille

## Créer une scène 3D interactive

# Tutoriel 1

## Ma première scène



Philippe Roy <[philippe.roy@ac-grenoble.fr](mailto:philippe.roy@ac-grenoble.fr)>

<https://forge.aeif.fr/blender-edutech/blender-edutech-tuto>

# Objectif



L'objectif de ce tutoriel est de créer une **scène animée et interactive**. Le support est le **labyrinthe à bille** ; le principe est de tourner (2 axes) le plateau afin d'amener la bille à l'arrivée. Ce tutoriel est une déclinaison pour UPBGE du projet n°1 du livre "[Créez vos propres jeux 3D comme les pros](#)" (Éditions Graziel) de Grégory Gossellin De Bénicourt.

Le tutoriel se décompose en 11 étapes :

- [1. Installer Blender/UPBGE](#)
- [2. Modéliser le plateau](#)
- [3. Gestion de la lumière et de la caméra](#)
- [4. Définir les matériaux](#)
- [5. Déplacer le plateau avec les briques logiques](#)
- [6. Créer la bille et définir sa physique](#)
- [7. Définir le gameplay : règle d'échec](#)
- [8. Modéliser la panneau de victoire](#)
- [9. Définir le gameplay : règle de victoire](#)
- [10. Fermer le panneau de victoire par clic](#)
- [11. Animer le panneau de victoire par des images-clés](#)
- [12. Produire un exécutable \(GNU/Linux, Windows, macOS\)](#)

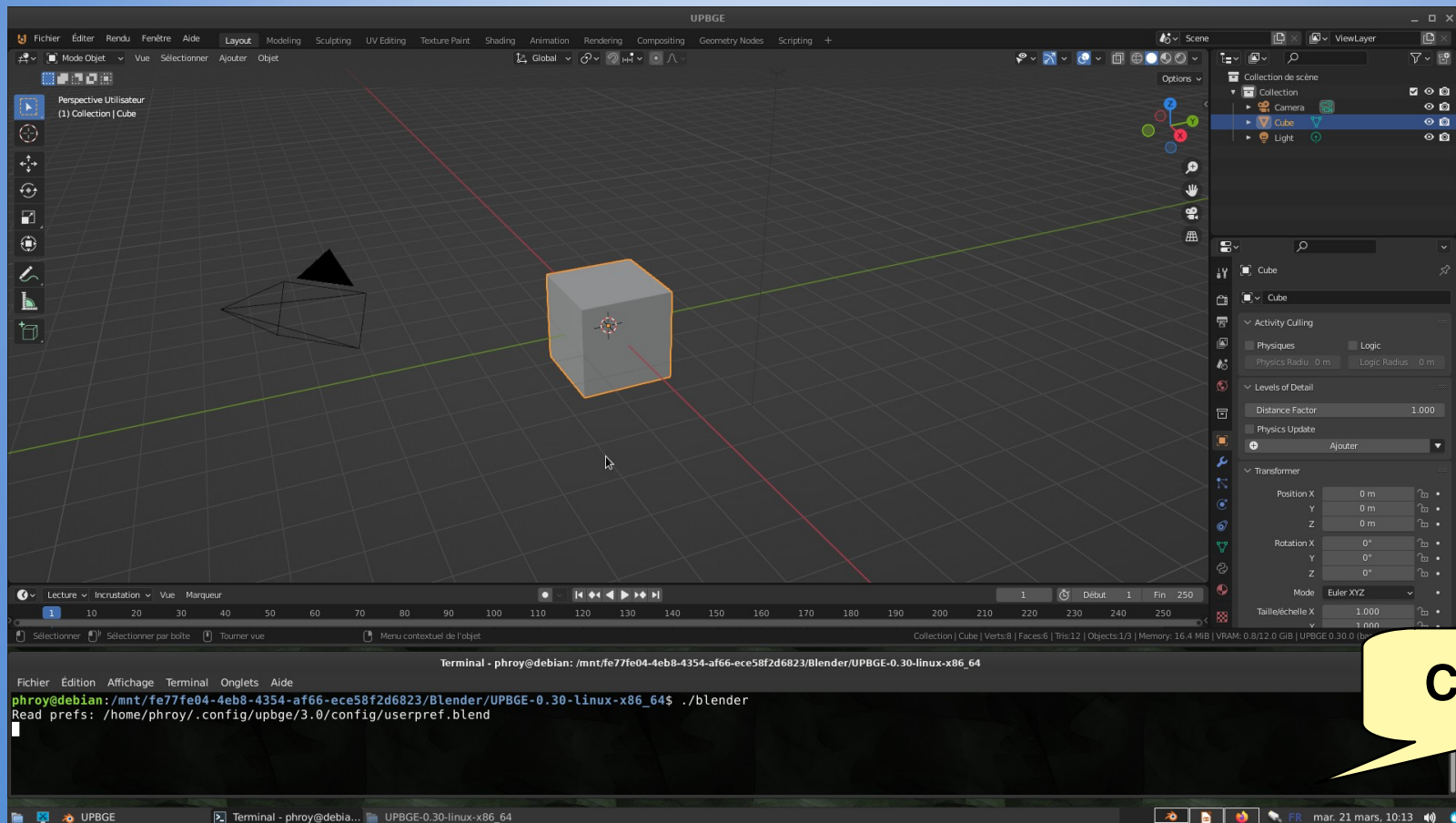


# 1. Installation de Blender/UPBGE



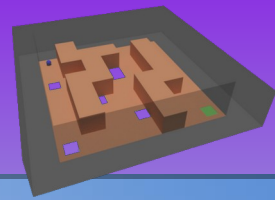
Le moteur de jeu **UPBGE** (**UP** Blender **G**ame **E**ngine) intègre déjà Blender, donc seule son installation suffit. Lancer UPBGE dans une **console** permet de visualiser les messages (erreurs, sortie standard, ...).

- **UPBGE** (version 0.3+) se trouve à cette adresse : <https://upbge.org>



Console

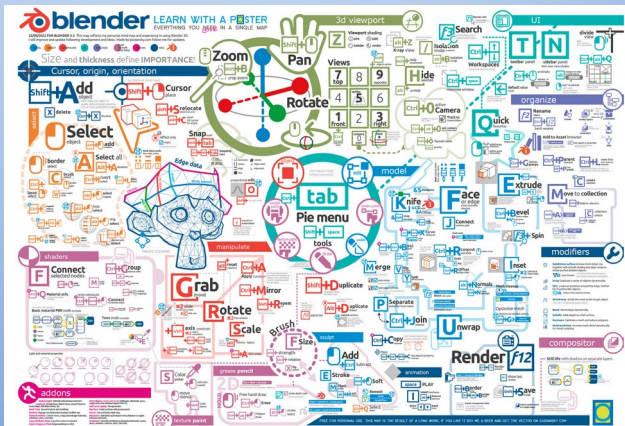
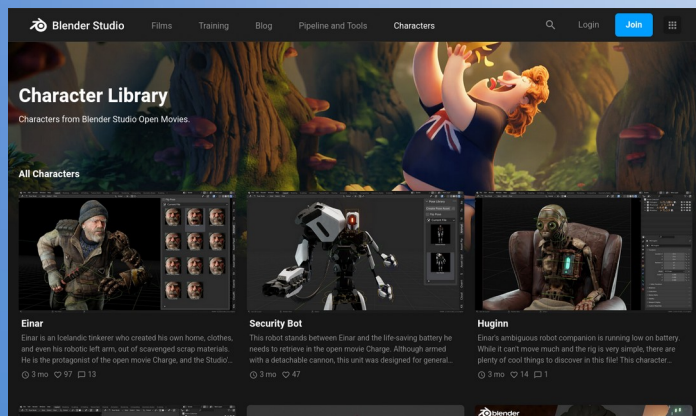
# 1. Installation de Blender/UPBGE



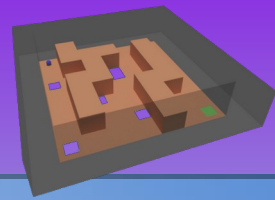
Ce tutoriel n'abordera pas dans les détails l'utilisation de Blender. La guidance ne nécessite pas de pré-requis et elle est autosuffisante. Par contre, elle sera donc très orientée vers les tâches liées aux étapes du tutoriel.

En terme de ressources générales :

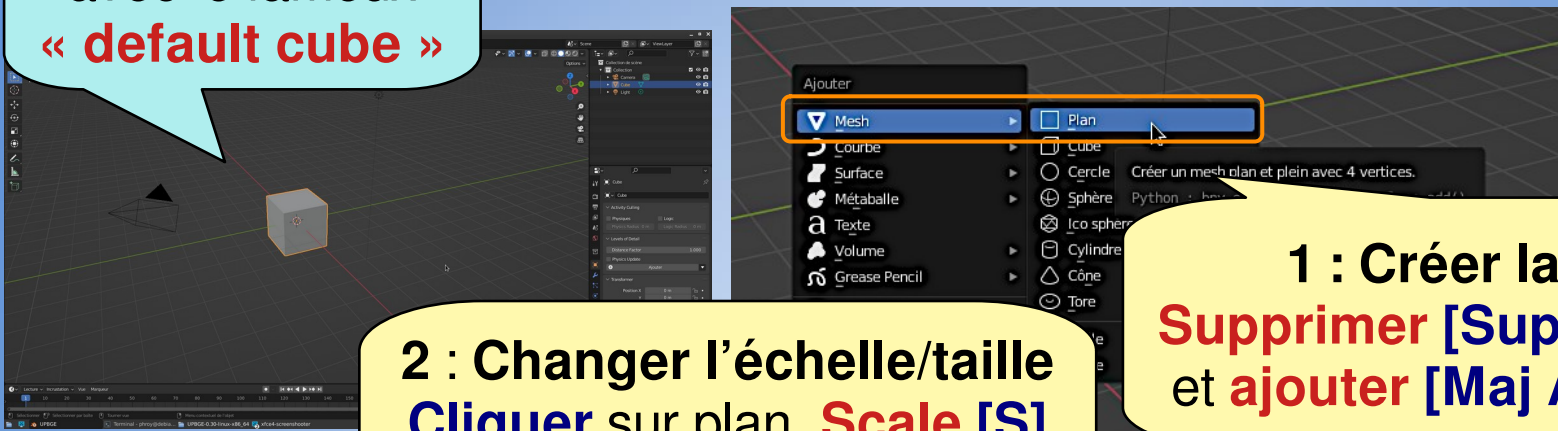
- le [Manuel officiel de Blender](#)
- le [Manuel officiel de UPBGE](#)
- le [studio de création des développeurs de Blender](#)
- le très bon livre "[La 3D libre avec Blender](#)" (Éditions Eyrolles) de Olivier Saraja, Henri Hebeisen et Boris Fauret.
- la [Blender map de Giuliano D'Angelo](#)
- Les forums : <https://blenderartists.org>, <https://www.blendernation.com/>, Discord UPBGE, Discord Blender France, ...



## 2. Modéliser le plateau

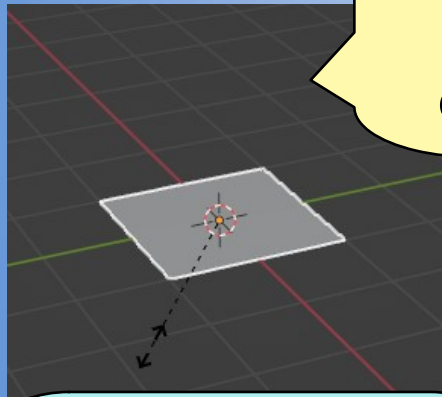


Scène de départ, avec le fameux « **default cube** »



**1 : Créer la base**  
**Supprimer [Suppr]** le cube et **ajouter [Maj A]** un **Plan**.

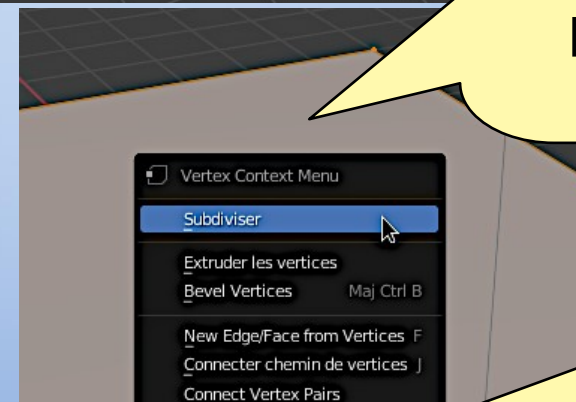
**2 : Changer l'échelle/taille**  
**Cliquer** sur plan, **Scale [S]** puis **saisir** le **facteur d'échelle** ici **5 [Entrée]**.



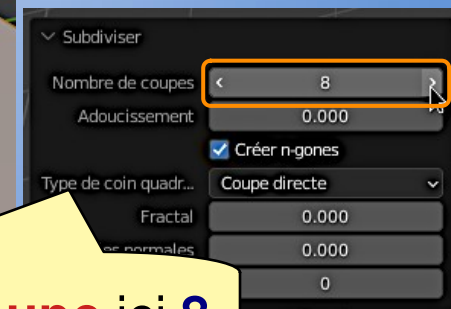
**3 : Affiner le plan**  
**Basculer** en **mode Édition [Tab]** puis **clic droit** et **Subdiviser**



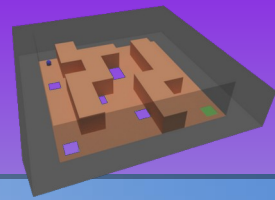
La **barre latérale** indique que le plan a une échelle de 5 et une dimension de 10x10m



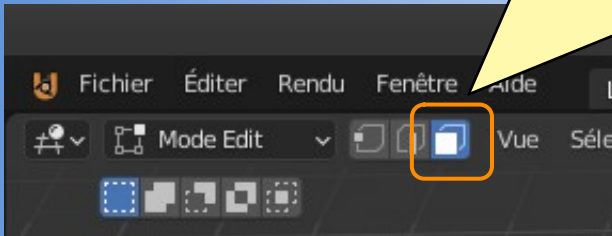
**Saisir** le **nombre de coupe** ici **8**.



## 2. Modéliser le plateau



**4 : Sélectionner que les faces**  
Toujours en **mode Édition**, cliquer sur le **Filtre de sélection face**.



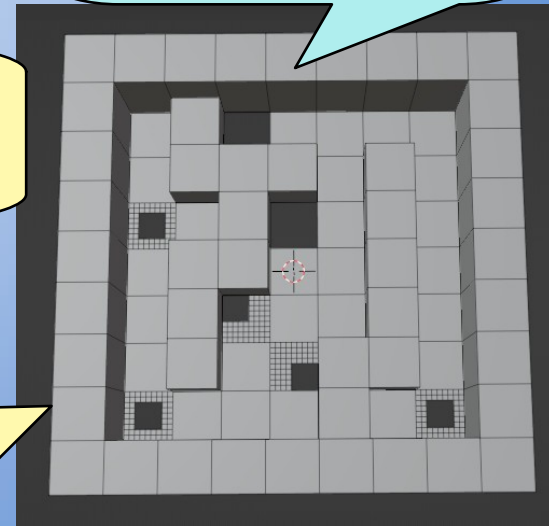
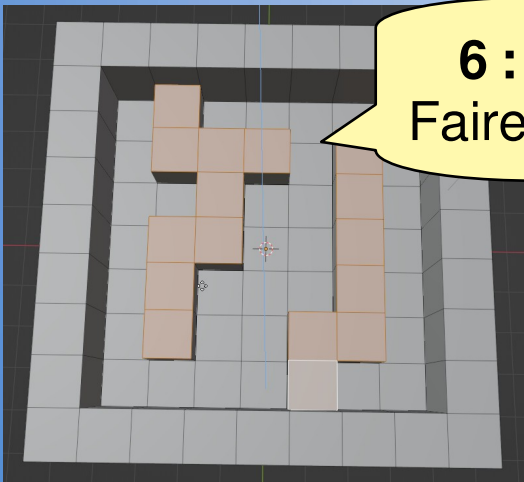
**[Maj] + Clic** pour ajouter à la sélection  
**[Ctrl] + Clic** permet une sélection par chemin le plus court

**5 : Extruder les murs extérieurs**  
**Cliquer** sur les faces des murs, **Extruder [E]**, puis saisir la **longueur de l'extrusion** ici **2 [Entrée]**.

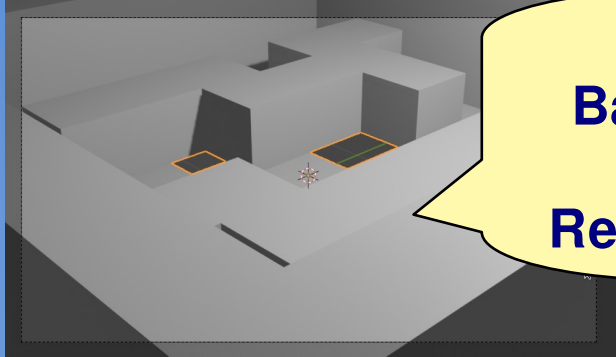
Subdiviser les faces pour avoir des trous plus petits

**6 : Extruder les murs intérieurs**  
Faire de même avec une hauteur de 1.

**7 : Faire les pièges et l'arrivée**  
**Supprimer les faces [Suppr]**.



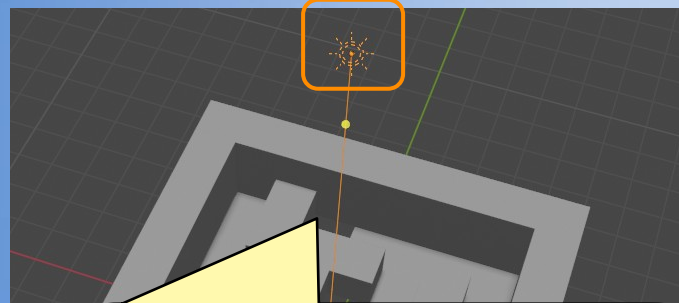
# 3. Gestion de la lumière et de la caméra



**1 : Afficher la vue de la caméra**  
**Basculer** vers la **vue caméra** [Numpad 0],  
pas vraiment adaptée au gameplay !  
**Revenir** à la **vue générale** avec [Numpad 0]



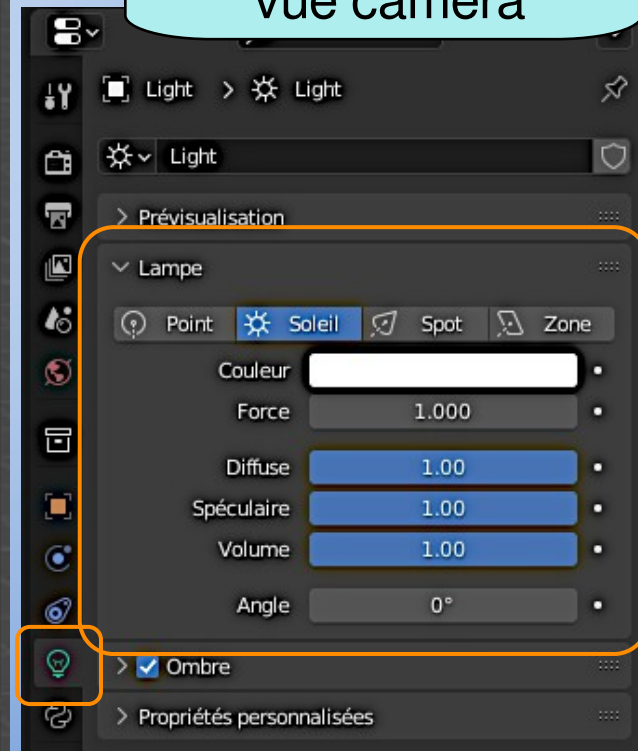
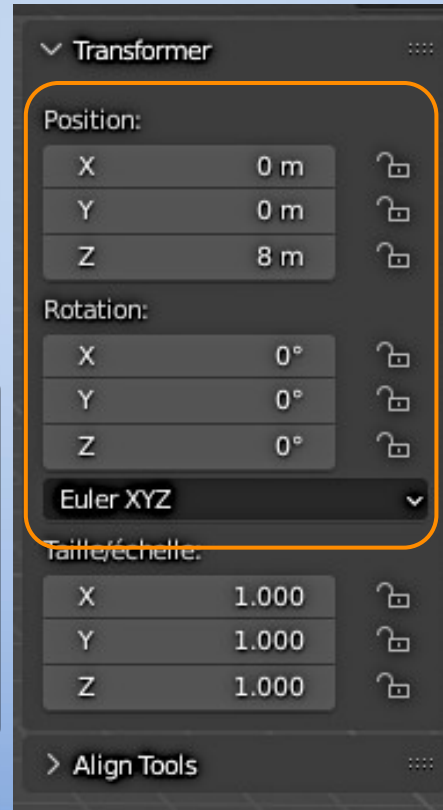
Bouton de bascule  
vue caméra



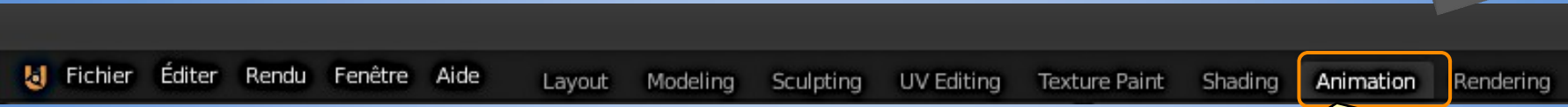
**2 : Définir, positionner et orienter la lumière**

Sélectionner la lumière, définir

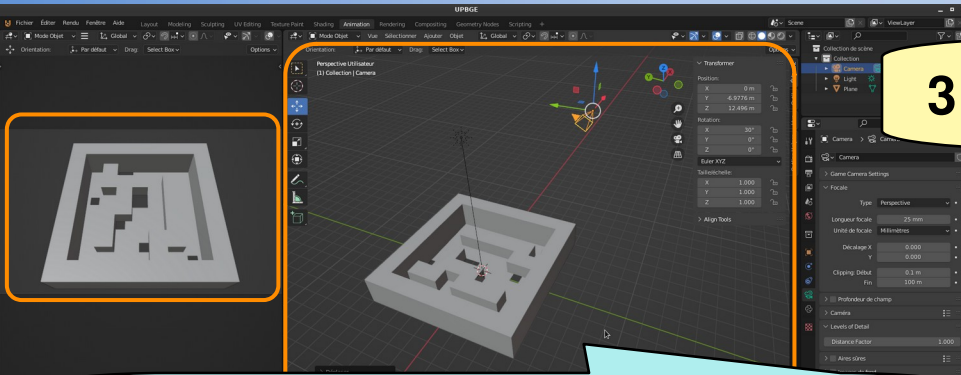
- son **type** sur **Soleil**
- sa **intensité** sur **1**
- sa **position** (x,y,z) sur **(0,0,0)**
- son **orientation** sur **(0,0,0)**



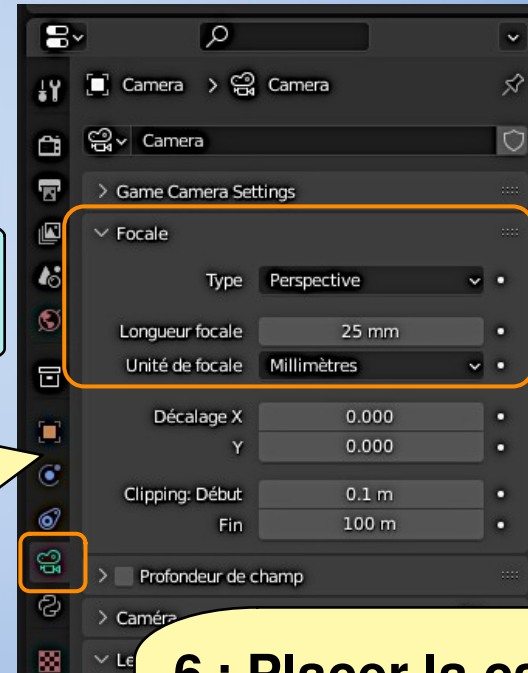
# 3. Gestion de la lumière et de la caméra



3 : Passer sur le bureau Animation



Ce bureau permet d'avoir la vue caméra et la vue de travail en même temps.

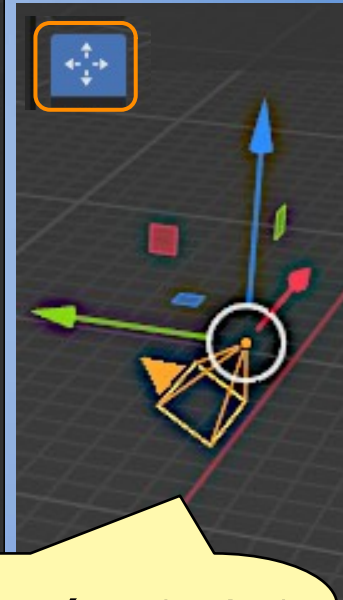


4 : Définir la focale  
Sélectionner la camera, définir sa focale sur 25mm

5 : Placer la caméra

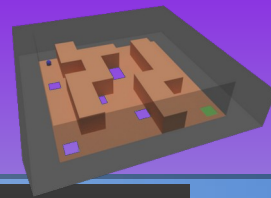
- sa position sur x sur 0
- son orientation sur x sur 30°

6 : Placer la caméra (suite)  
Choisir l'outil déplacer puis avec le trièdre déplacer la caméra sur chaque axe.





# 4. Définir les matériaux



Fichier Éditer Rendu Fenêtre Aide Layout Modeling Sculpting UV Editing Texture Paint **Shading** Animation Rendering

2 : Créer les deux matériaux « Brun » et « Transparent »

1 : Passer sur le bureau Shading

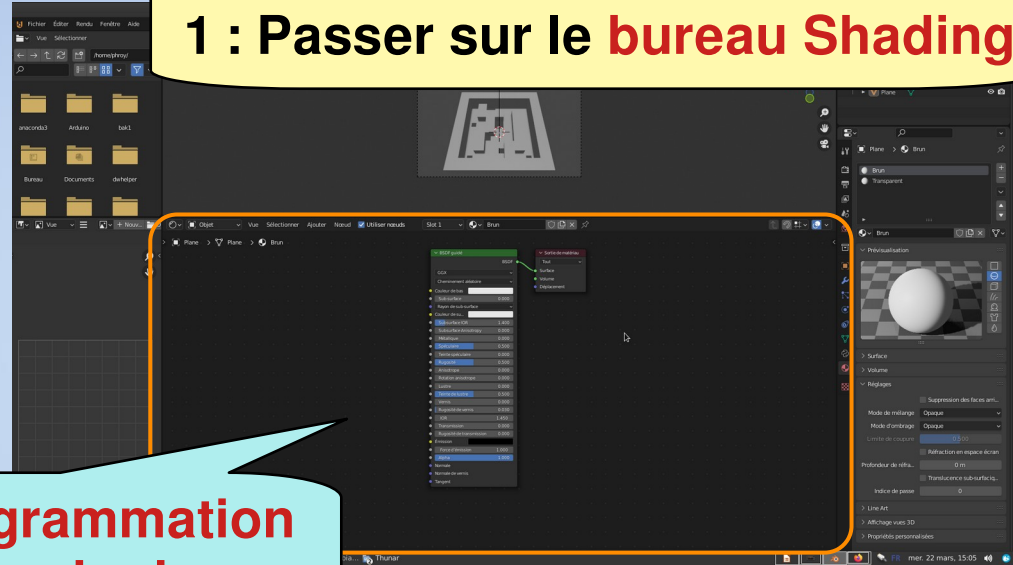
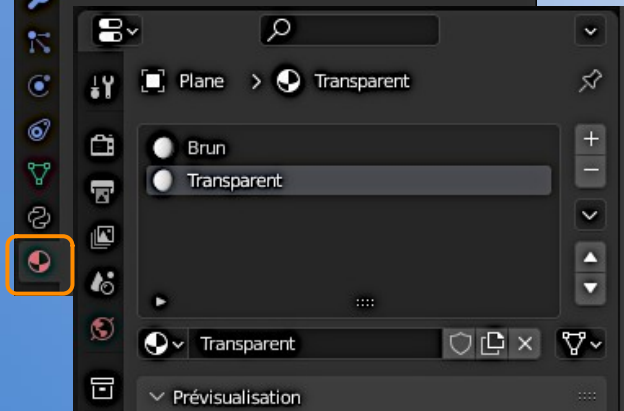
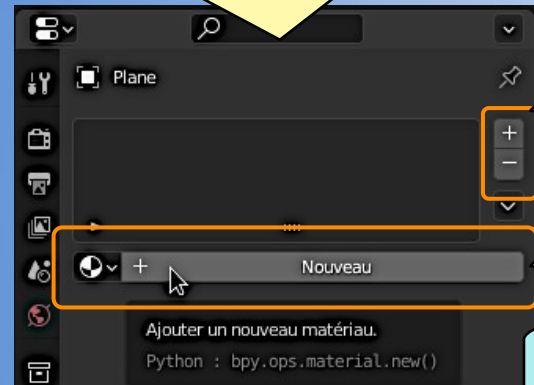
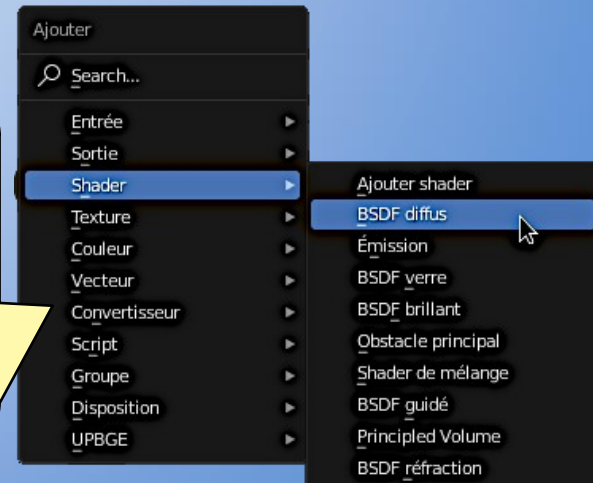
Ajouter un slot

Ajouter un matériau

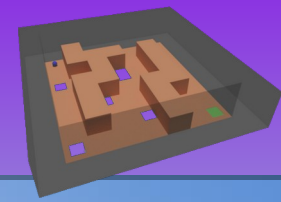
Zone de programmation nodale des shaders

3 : Définir le matériau « Brun »

- **Supprimer [Suppr]** la node **BSDF guidé**
- **ajouter [Maj A]** la node **Shader BSDF diffus**



# 4. Définir les matériaux



Color picker showing RGB, HSV, and Hex values. Below it, a material node labeled 'Sortie de matériau' with a dropdown menu set to 'Tout'. To the left, a 'BSDF diffus' node with 'Couleur' set to orange, 'Rugosité' at 0.000, and 'Normale' selected.

4 : Définir le matériau « Brun » (suite)

Définir la **couleur** avec la **palette** (cliquer sur la couleur) et **connecter** le **shader** avec la **sortie** (surface)

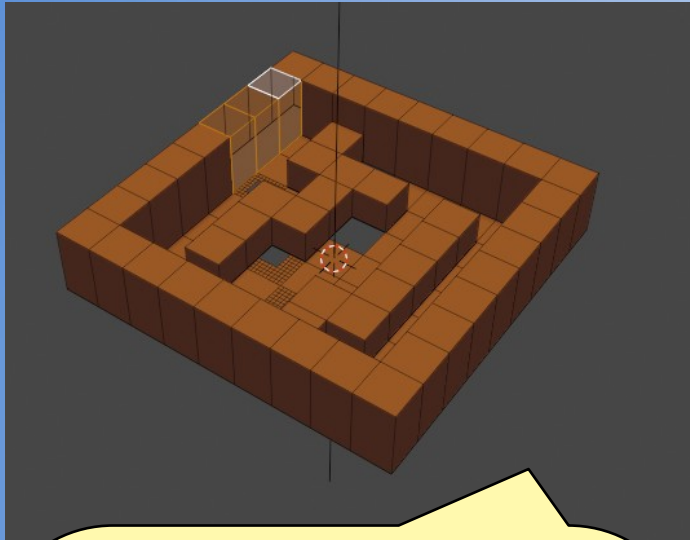
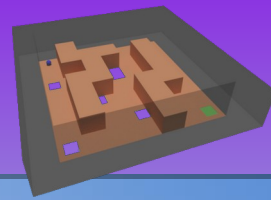
The material editor shows 'Brun' selected in the material list. The preview window shows a sphere with a brown color. The 'Réglages' panel is partially visible at the bottom.

5 : Définir le matériau « Transparent »

Faire de même, avec le **shader BSDF Transparence**, la **couleur grise** (valeur 0.5) et le **mélange** sur **Alpha mélangé** et **sans ombrage**

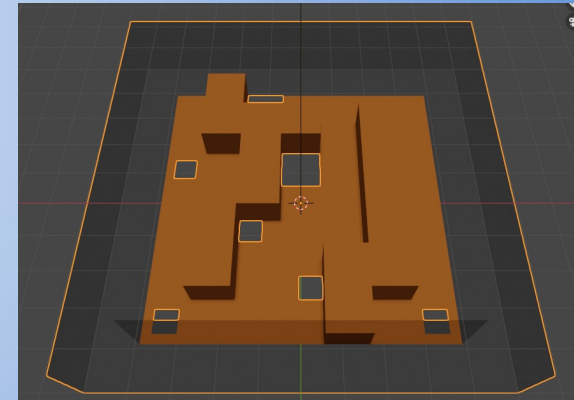
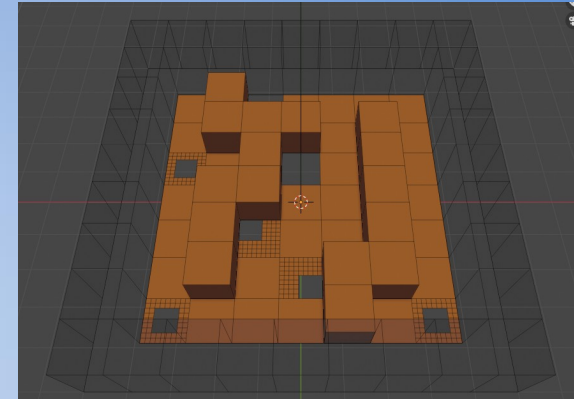
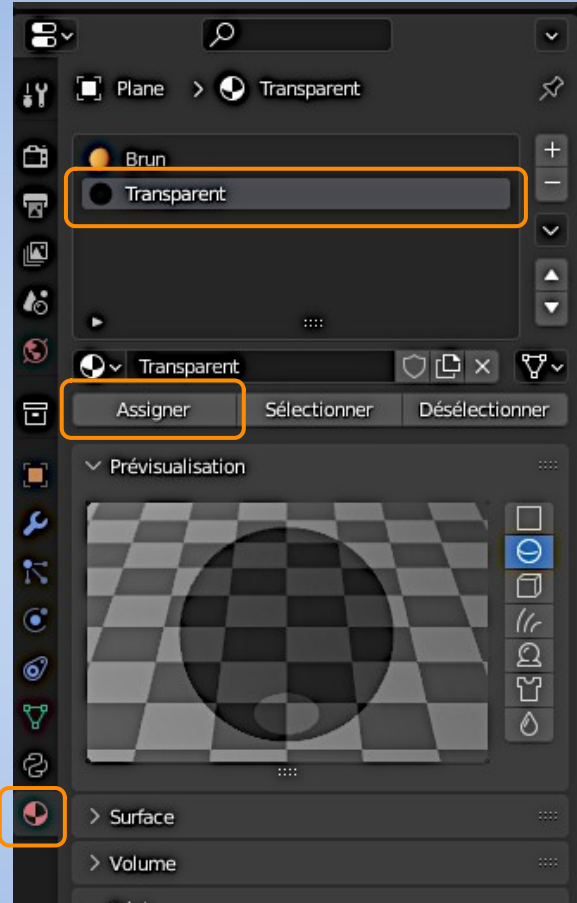
The material editor shows 'Transparent' selected. The preview window shows a sphere with a transparent effect. The 'Réglages' panel is expanded, showing 'Mode de mélange' set to 'Alpha mélangé', 'Mode d'ombrage' set to 'Aucun', and 'Limite de coupure' set to 0.500.

# 4. Définir les matériaux



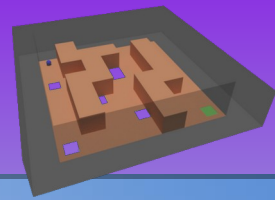
## 6 : Rendre les murs extérieurs transparents

- En **mode Édition [Tab]** sélectionner les **faces** des murs extérieurs,
- puis **assigner** le matériau «**Transparent**» aux faces.

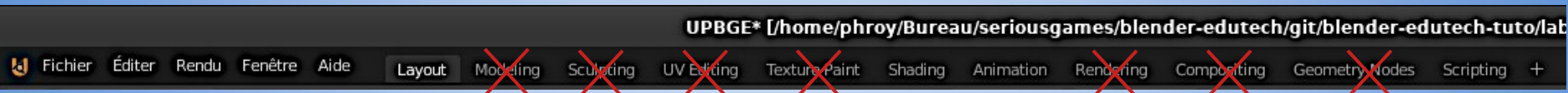


Plateau final **vue caméra [Numpad 0]**, en **mode Édition** et en **mode Objet [Tab]**, Alors ce rendu ?

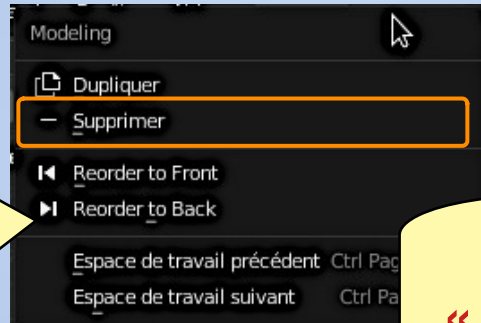
# 5. Déplacer le plateau avec les briques logiques



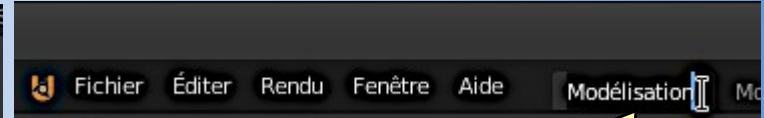
Le **bureau** de programmation avec les **briques logiques UPBGE** (logic brick) n'est installé par défaut. Nous allons profiter de cette étape pour réorganiser nos bureaux.



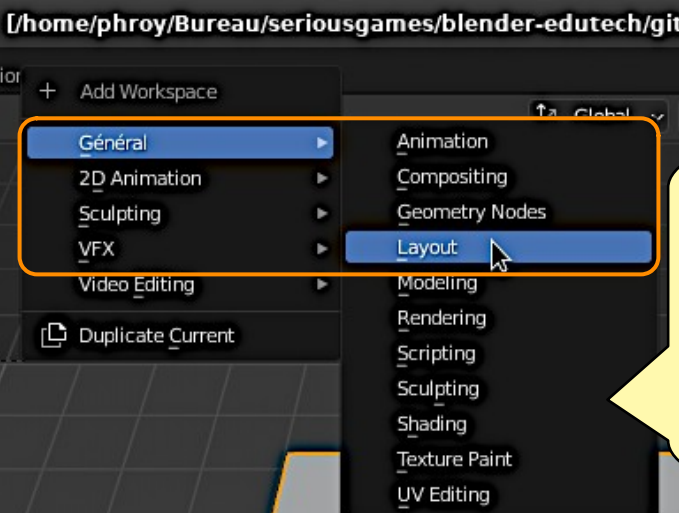
**1 : Enlever les bureaux non utilisés**  
**Clic droit** sur l'onglet puis **Supprimer**



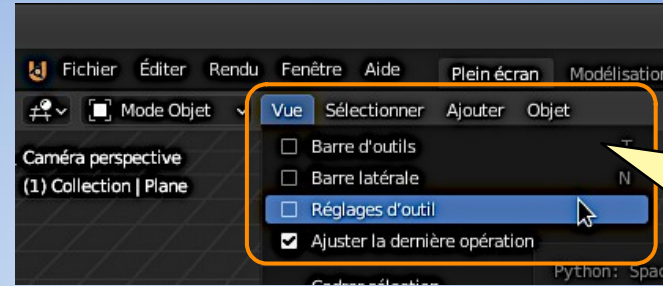
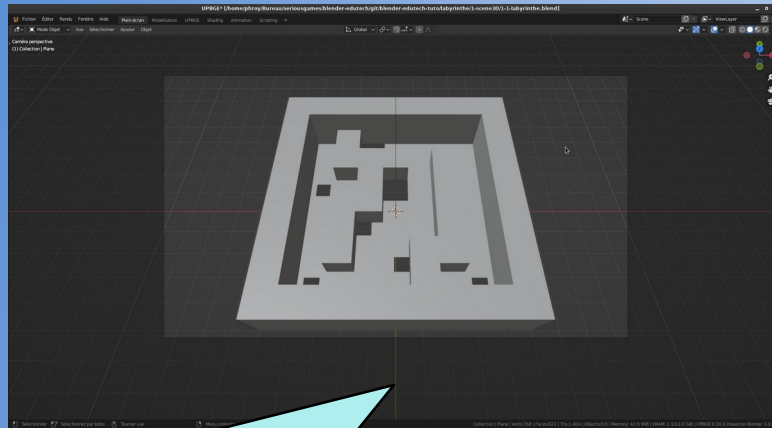
**2 : Renommer le bureau « Layout » en « Modélisation »**  
**double clic** sur l'onglet



- 3 : Créer le bureau Plein écran**
- **Ajouter** un bureau avec le **+** (basé sur **Layout**),
  - le placer en début **clic droit** sur l'onglet et **Reorder to Front**,
  - puis renommer le.



# 5. Déplacer le plateau avec les briques logiques

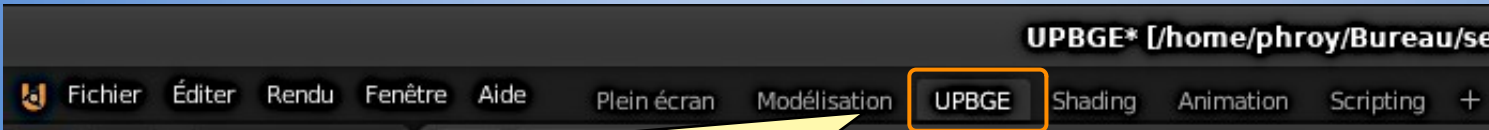


4 : Cacher les barres  
**Menu Vue**



5 : Joindre les zones pour ne faire qu'une  
**Clic droit** sur la **jointure** entre deux zones,  
**Joindre les zones** puis indiquer la zone restante.

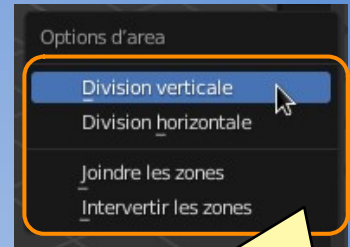
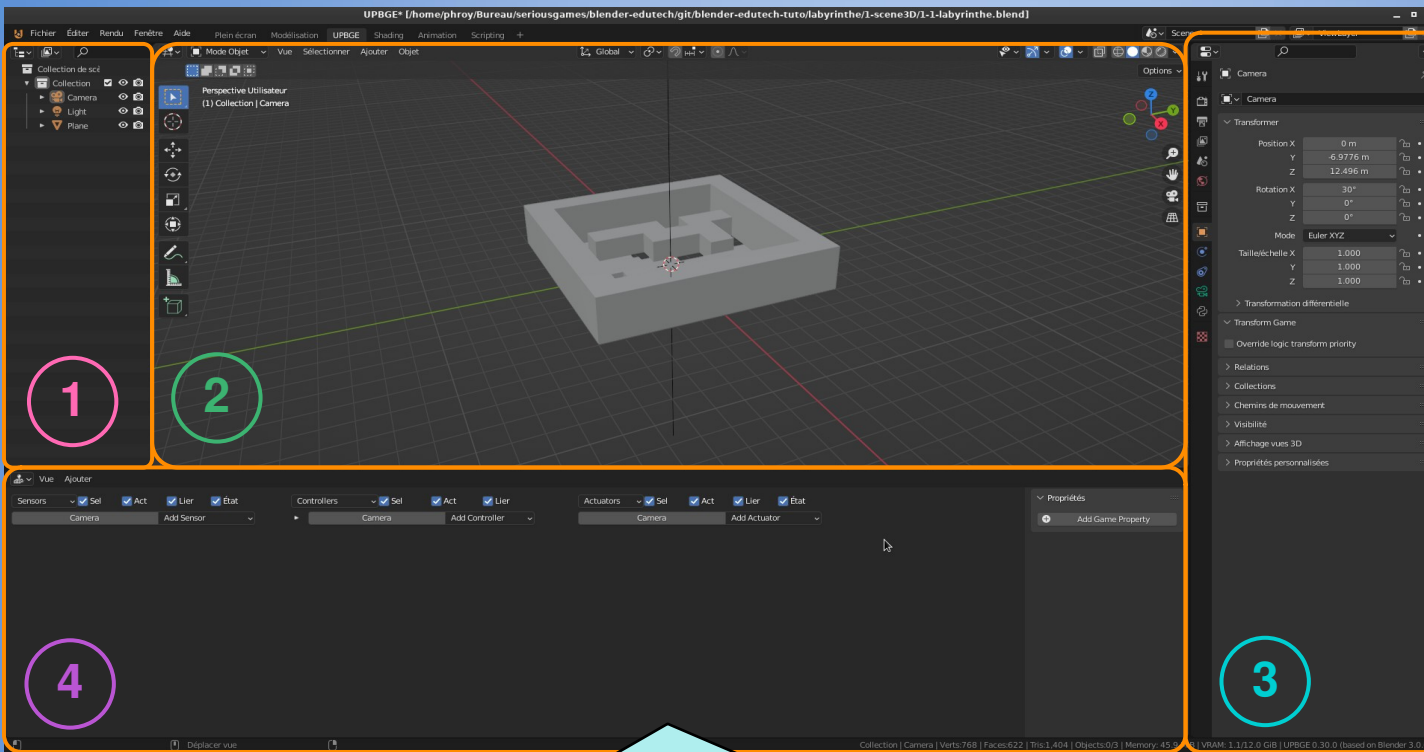
Le bureau **Plein écran** va servir à exécuter le jeu dans **Blender**, c'est donc la **vue Caméra** dépourvue de toutes autres **zones**.



## 6 : Créer le bureau UPBGE

De la même manière créer le bureau **UPBGE** (aussi à partir de **Layout**)

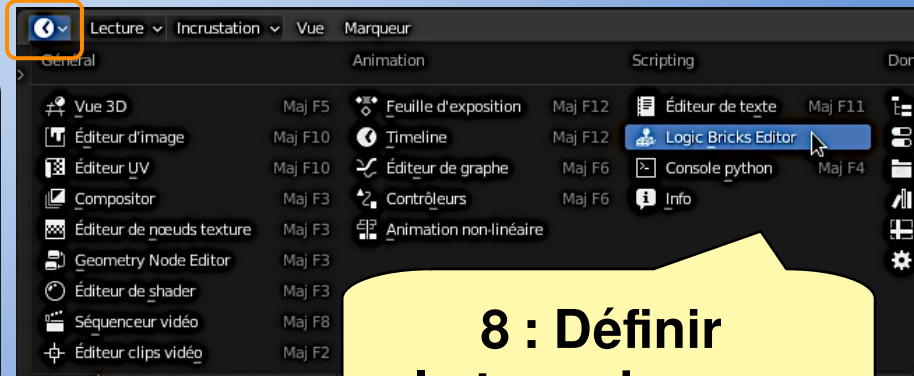
# 5. Déplacer le plateau avec les briques logiques



**7 : Joindre / Créer des zones**  
**Clic droit** sur la **jointure** entre deux zones.

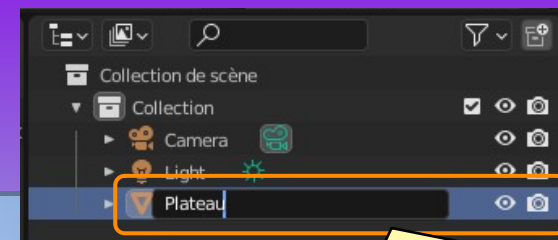
Le bureau **UPBGE** se décompose en **4 zones** :

- **1** : Arbre des objets (**Outliner**)
- **2** : **Vue 3D**
- **3** : Panneau des **Propriétés**
- **4** : **Éditeur de briques logiques**



**8 : Définir le type de zone**

# 5. Déplacer le plateau avec les briques logiques



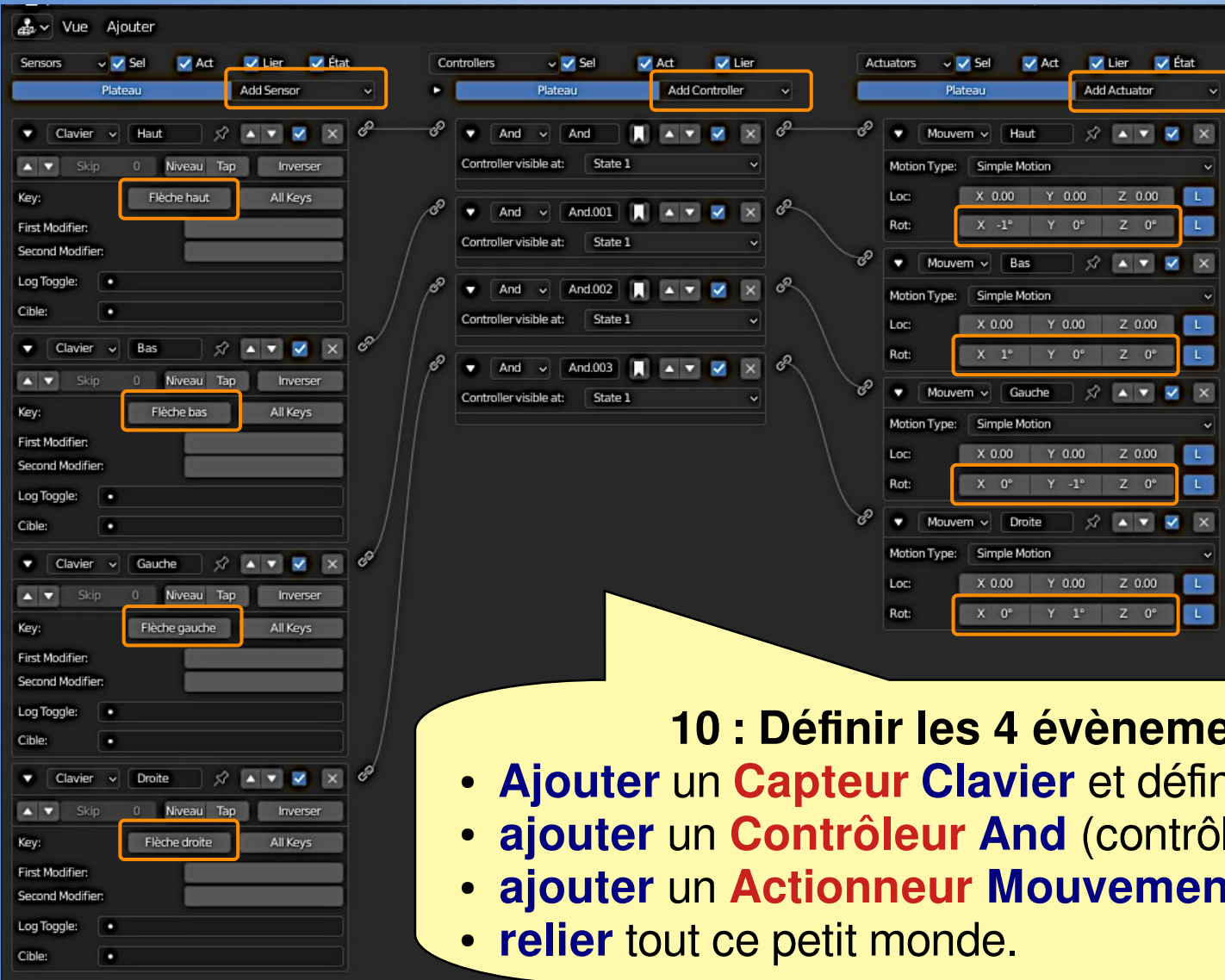
9 : Renommer l'objet avec « Plateau »

-1° en X pour Flèche haut

1° en X pour Flèche bas

-1° en Y pour Flèche gauche

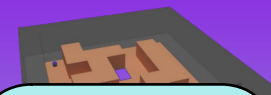
1° en X pour Flèche droite



## 10 : Définir les 4 évènements

- Ajouter un **Capteur Clavier** et définir la **touche**,
- ajouter un **Contrôleur And** (contrôleur standard),
- ajouter un **Actionneur Mouvement** avec l'**angle**
- **relier** tout ce petit monde.

# 5. Déplacer le plateau avec les briques logiques

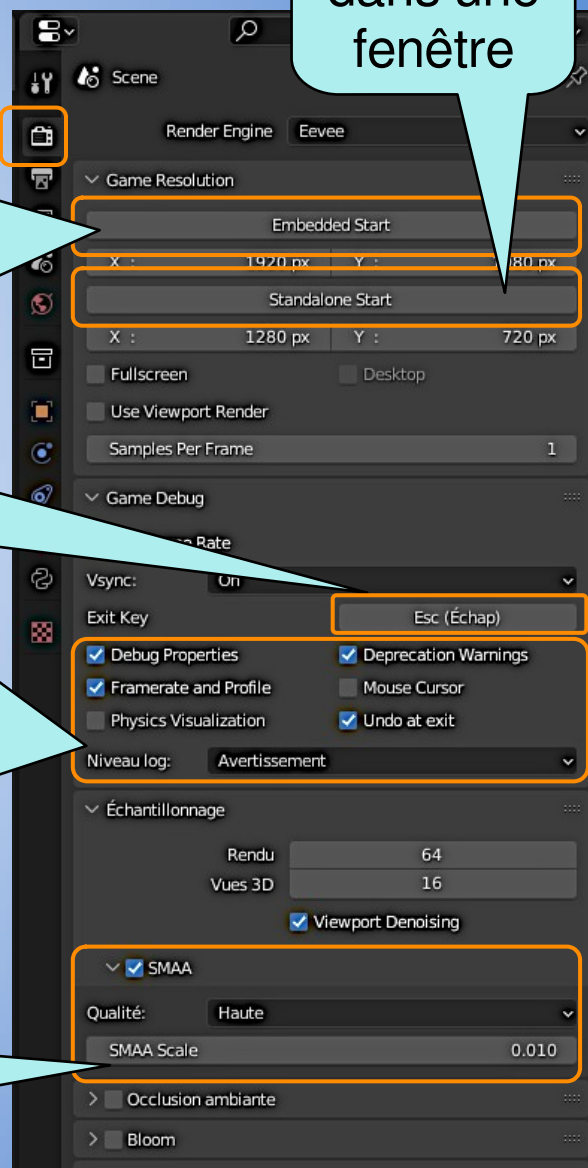
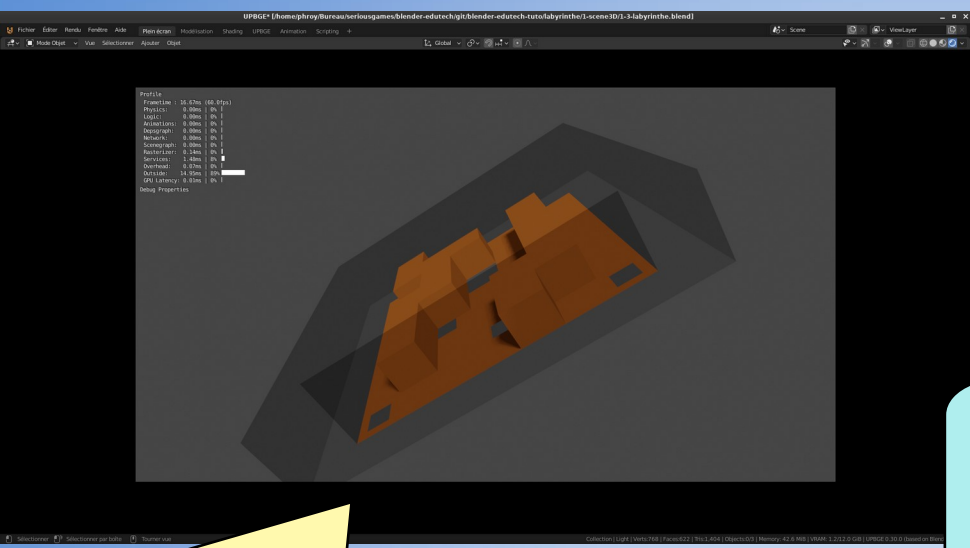


Rendu dans une fenêtre

Rendu dans Blender comme [P]

Sortir du rendu avec [Esc]

Affiche  
• les propriétés des objets 3D  
• les durées de traitement (Framerate)



11 : Tester le programme

- Aller sur le bureau **Plein écran**,
- lancer le moteur de jeu par [P]

Si si le plateau tourne bien ...  
Le Blender interactif commence maintenant !

Défini la qualité de l'anti-crénelage

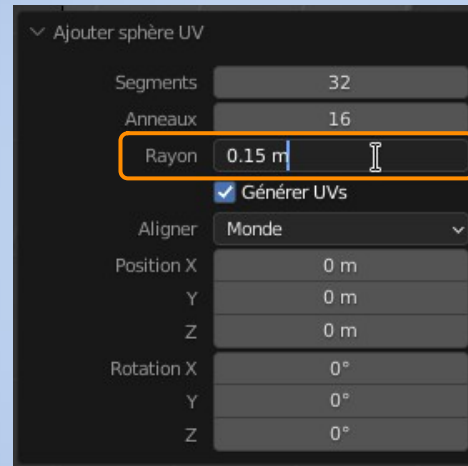
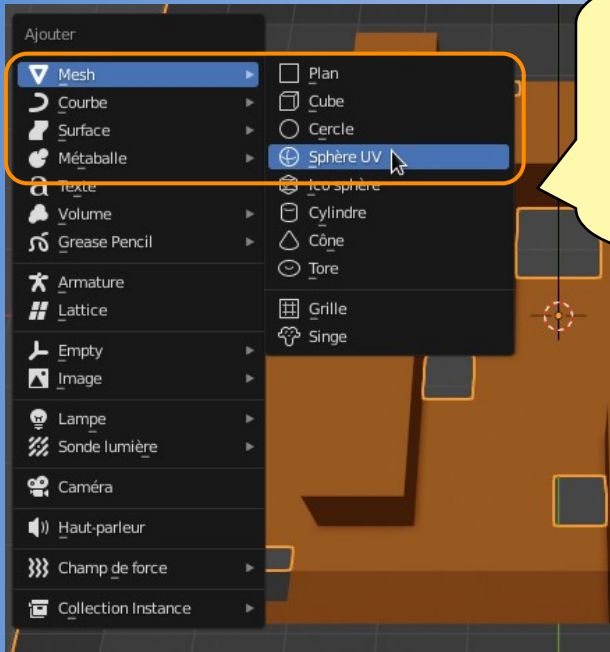


# 6. Créer la bille et définir sa physique

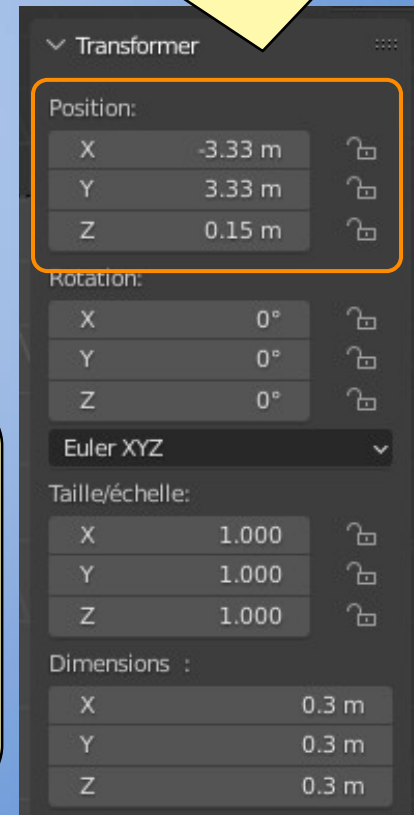
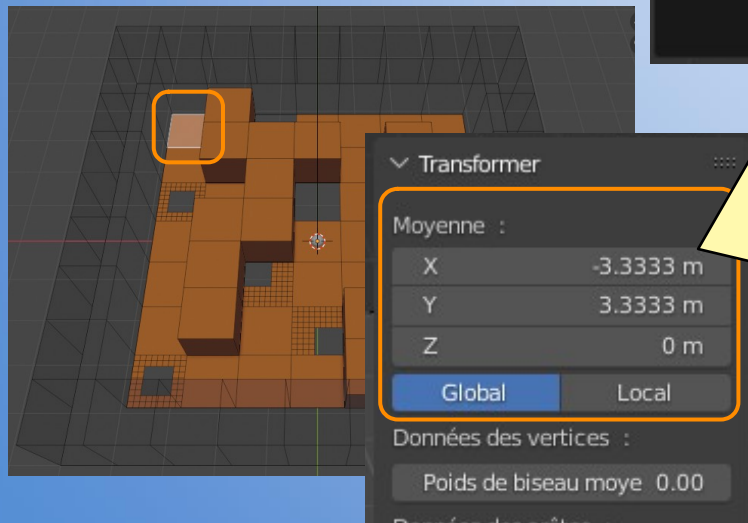


**1 : Créer la bille**  
Ajouter [Maj A] une **Sphère UV** puis saisir son **rayon 0.15 m**

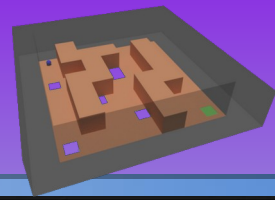
**3 : Positionner la bille**  
En **mode Objet [Tab]**, cliquer sur la **bille** et saisir sa **position (-3.33, 3.33, 0.15)**



**2 : Localiser le point de départ de bille**  
En **mode Edition [Tab]**, cliquer sur la **face de départ** et relever les **coordonnées globales** de son **centre**



# 6. Créer la bille et définir sa physique



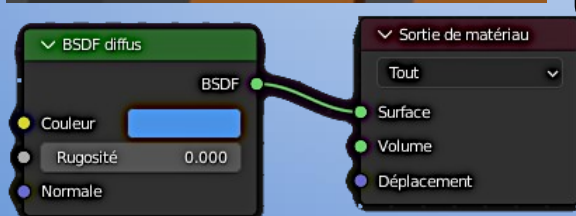
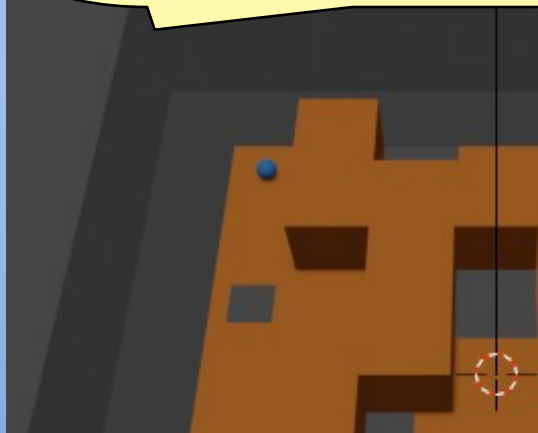
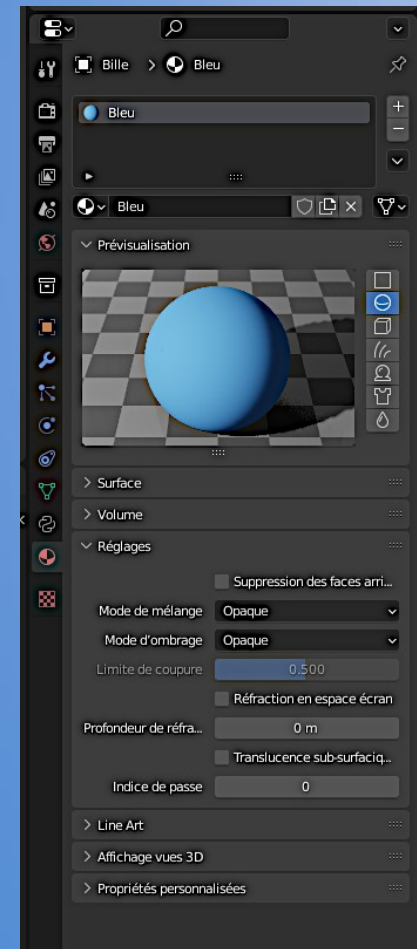
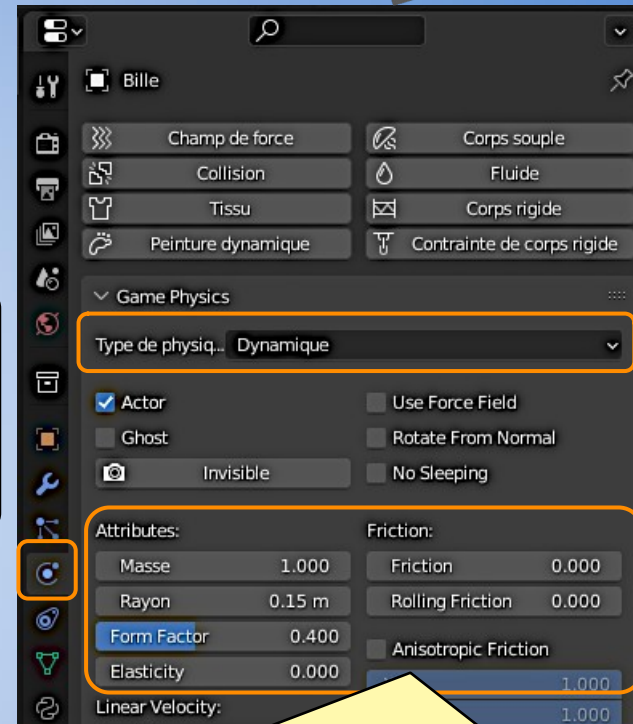
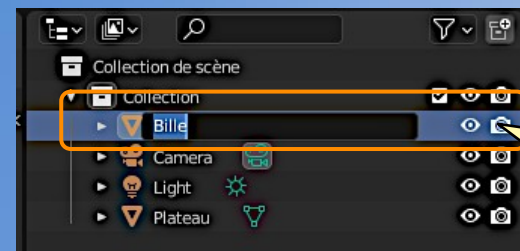
4 : Renommer l'objet avec « Bille »

5 : Définir la **matière** de la **bille**  
Nous prendrons le **Shader diffus** avec une **couleur Bleu**

6 : Définir la **physique** de la **bille**

- **Type** : **Dynamique**
- **Rayon** : **0.15 m** (sa dimension)
- **Pas de friction**

7 : Tester la scène [P]

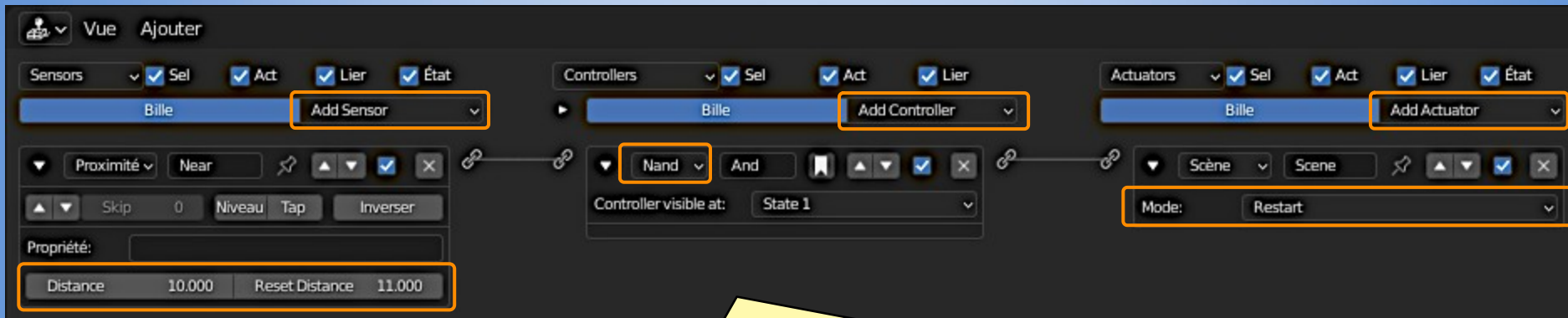


# 7. Définir le gameplay : règle d'échec



Le **gameplay** a deux règles :

- redémarrer la partie si la bille tombe dans un des pièges,
- afficher une fenêtre de victoire si la bille arrive à l'arrivé (codée à l'étape 9).

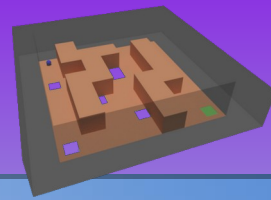


**1 : Détecter la proximité de la bille et du plateau, si la réponse est False alors redémarrer la partie**

- **Ajouter** un **Capteur Proximité** (near),
- définir la distance limite de détection à **10 m**,
- définir la distance d'annulation à **11 m**,
- **ajouter** un **Contrôleur Nand** (non et, inverseur),
- **ajouter** un **Actionneur Scène avec Restart**

**2 : Valider le redémarrage**  
**[P]**

# 8. Modéliser le panneau de victoire



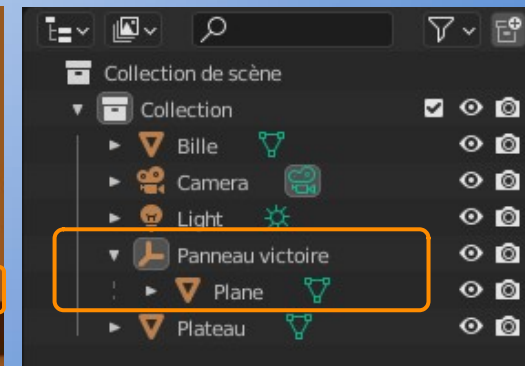
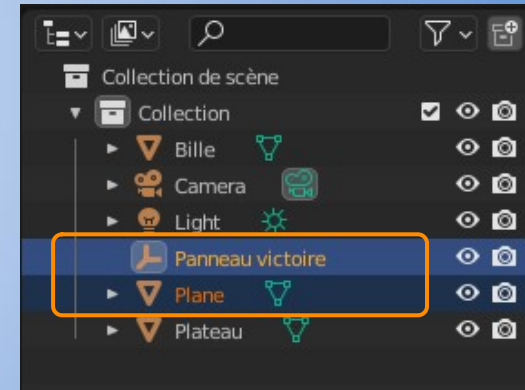
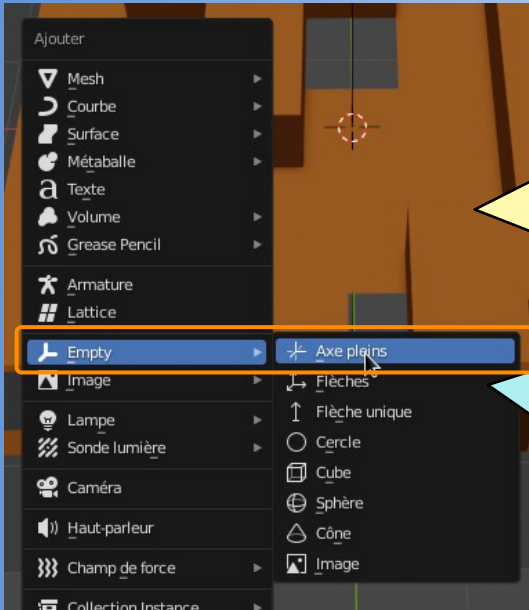
Le panneau de victoire apparaît en cas de réussite et affiche le temps écoulé et le nombre de tentative depuis la dernière réussite.

**1 : Créer un ensemble**  
**Ajouter [Maj A]** un objet  
**Axe pleins**, puis **renommer**  
le avec **Panneau victoire**

**2 : Créer le fond**  
**Ajouter [Maj A]**  
un **Plan**.

L'objet **Vide** (qui s'apparente  
ici à un repère) est comme  
un container pour grouper  
plusieurs objets.

**3 : Parenter le plan avec**  
**Panneau victoire**  
**Cliquer le plan (enfant),**  
**[Maj] clic sur le repère (parent),**  
puis **[Ctrl P] Objet**  
**(conserver transformations)**



# 8. Modéliser le panneau de victoire

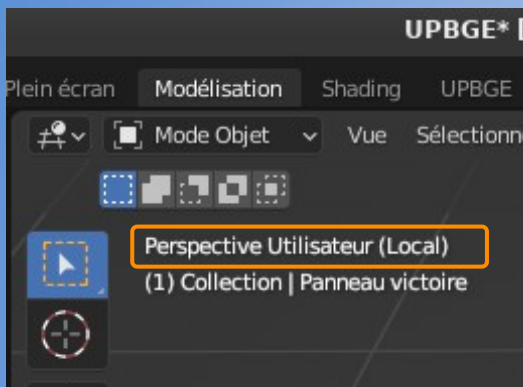
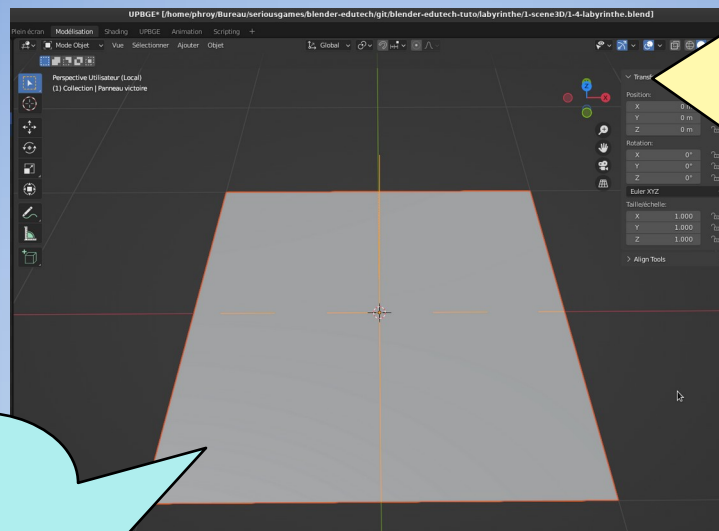
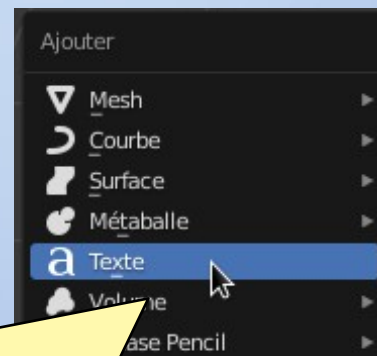
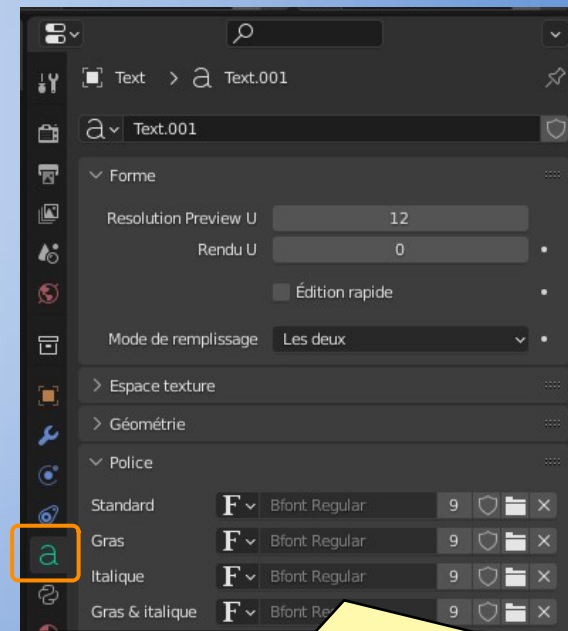


**4 : Se mettre dans la **Vue locale** du panneau**  
Sélectionner le repère **Panneau victoire** et le plan puis **[Numpad /]**.

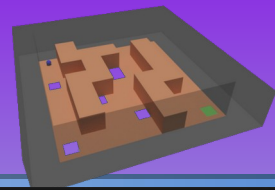
La **vue locale** sert à cacher les objets inutiles à l'instant t. C'est très utilisé lors de la modélisation des objets complexes.

- 5 : Ajouter le texte du Panneau**
- **Ajouter [Maj A] Texte**,
  - **saisir** le texte avec le **mode Edition [Tab]**,
  - **positionner** le.

**6 : Paramétrer le texte**  
voir page suivante



# 8. Modéliser le panneau de victoire



## « Victoire »

- Position (x,y,z) : **(0, 0.4, 0)**
- Extruder : **0.01**
- Biseau : **0.01**
- Taille (Police) : **0.4**
- Alignement : **Centre**
- Couleur : **Noir**

## 6 : Paramétrer le texte (suite)

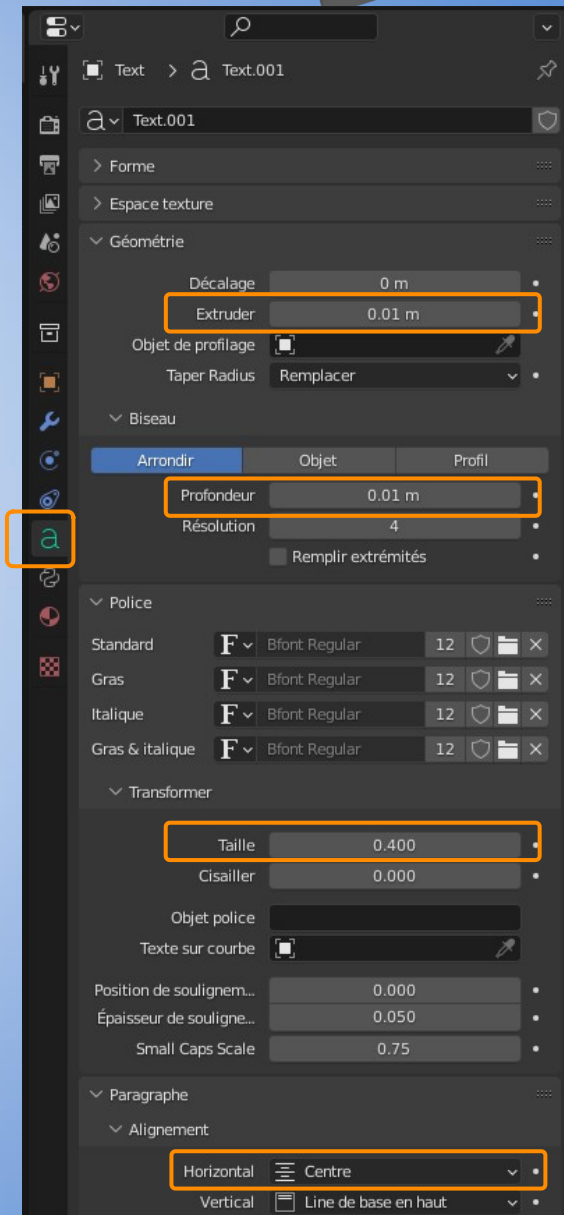
Le panneau comporte **2** objets **Texte**

## « Cliquer pour fermer »

- Position : **(0, -0.8, 0)**
- Extruder : **0.01**
- Biseau : **0.0025**
- Taille (Police) : **0.2**
- Alignement : **Centre**
- Couleur : **Noir**

**Victoire !**

**Cliquer pour fermer**



# 8. Modéliser le panneau de victoire

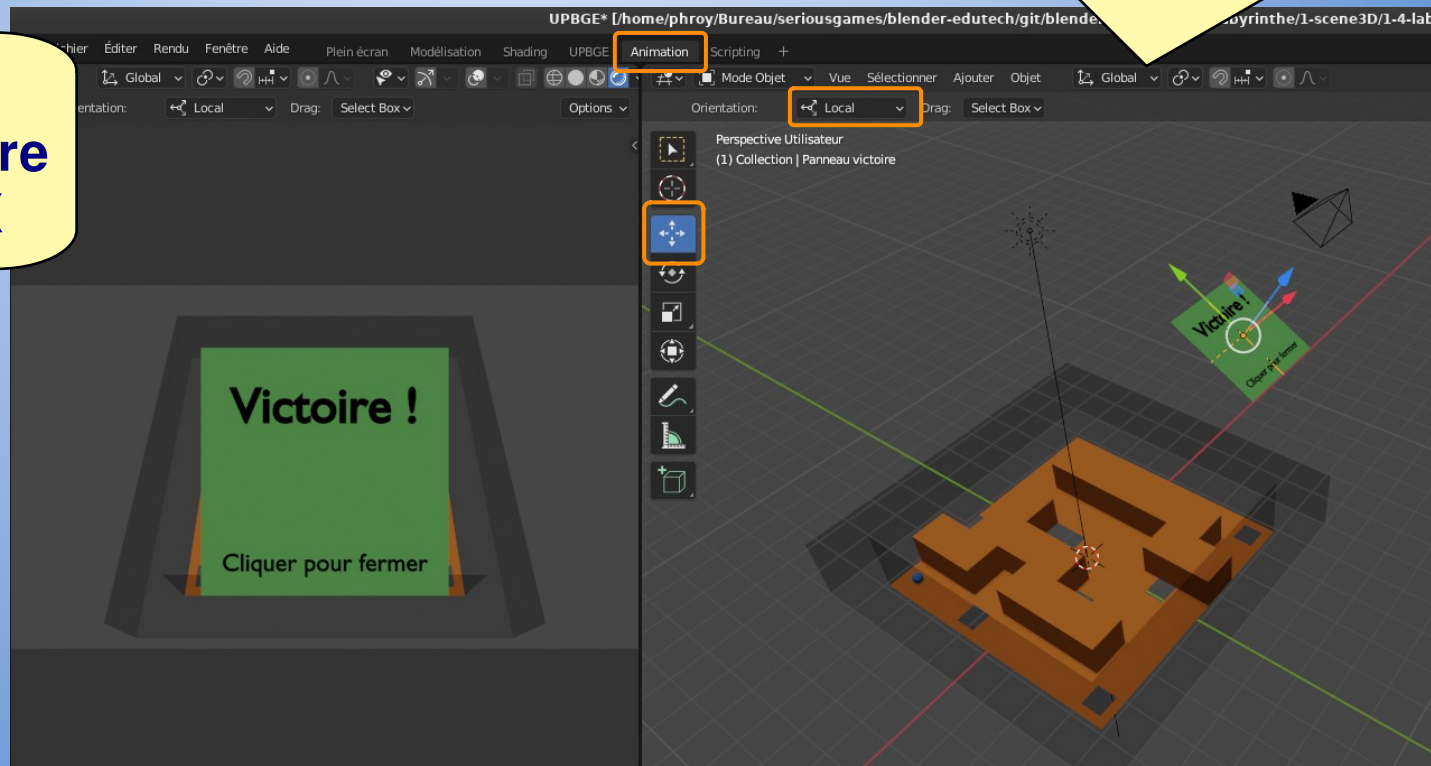
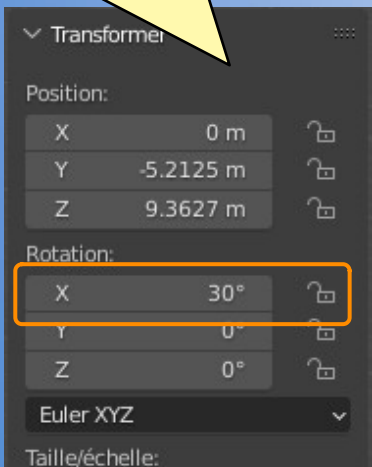


**7 : Parenter** les objets **Texte** avec **Panneau victoire** et **renommer** le plan avec **Panneau victoire - plan**

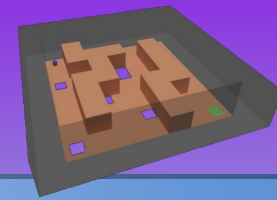
**8 : Définir la matière du plan**  
Nous prendrons le **Shader diffus** avec une **couleur Verte**

**10 : Positionner Panneau victoire**  
Sur le **bureau Animation**, **ajuster** la position avec le **Trièdre Local**

**9 : Orienter**  
**Panneau victoire**  
avec **30° en X**



# 9. Définir le gameplay : règle de victoire

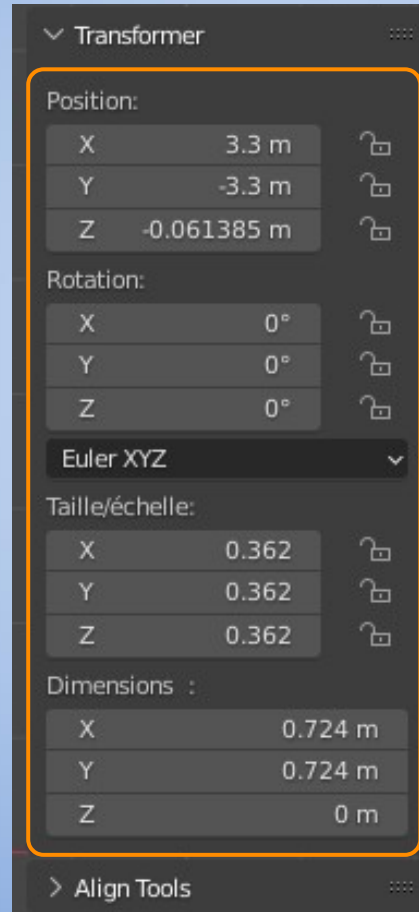


Pour détecter la présence de la bille à l'arrivée, nous allons donc mettre un plan qui sert déclencheur lors de sa collision avec la bille.

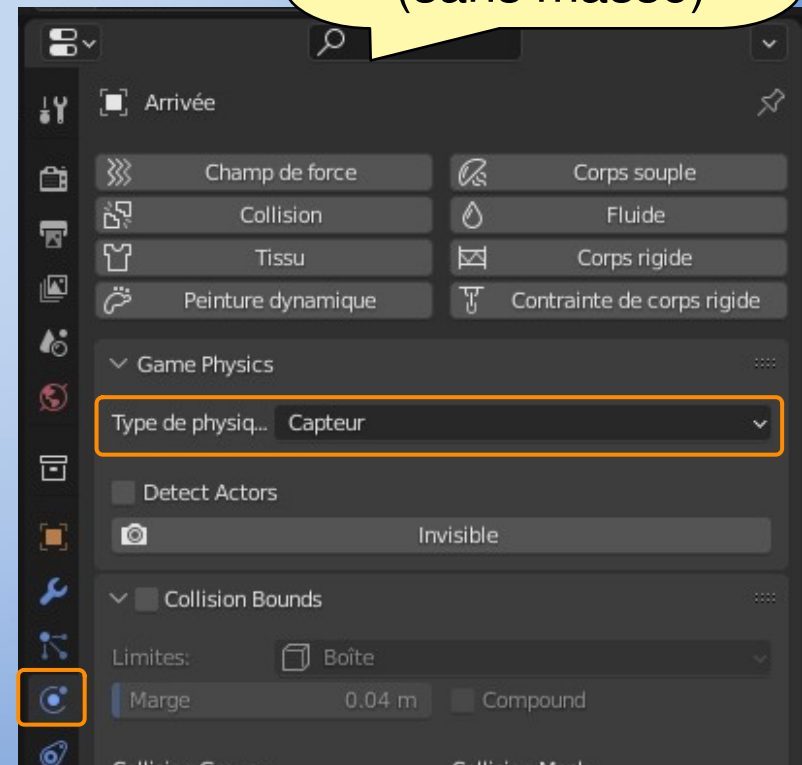


## 1 : Créer le plan d'arrivée

- **Nom** : Arrivée
- **Position** : sous l'arrivée
- **Couleur** : Verte

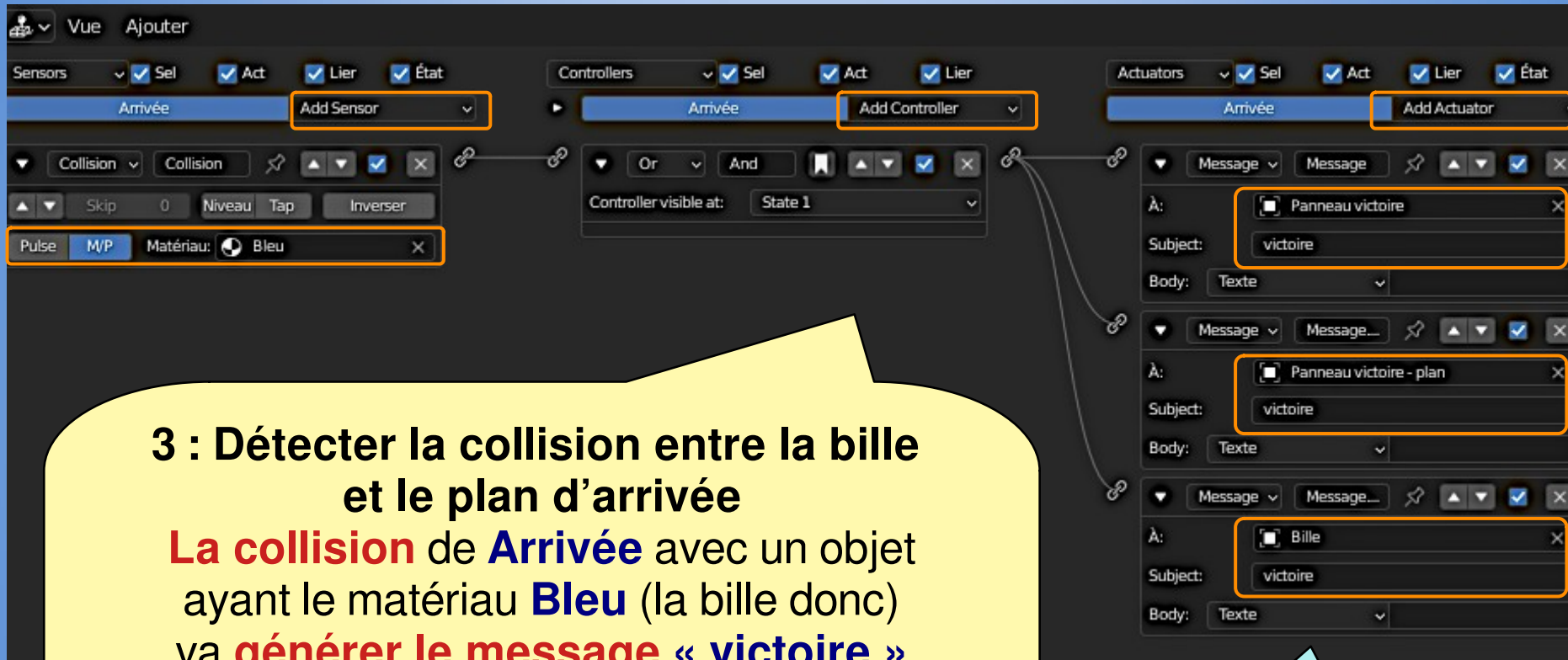


2 : Définir la physique du plan  
**Type** : Capteur  
(sans masse)





# 9. Définir le gameplay : règle de victoire



## 3 : Détecter la collision entre la bille et le plan d'arrivée

La **collision** de **Arrivée** avec un objet  
ayant le matériau **Bleu** (la bille donc)  
va **générer le message « victoire »**  
vers les trois objets :

- **Panneau victoire**
- **Panneau victoire – plan**
- **Bille**

Briques logiques  
de **Arrivée**

# 10. Fermer le panneau de victoire par clic



## 1 : Cacher le panneau de victoire en début de jeu

- **Ajouter** un **Capteur Toujours**,
- **ajouter** un **Contrôleur And**,
- **ajouter** un **Actionneur Visibilité** avec **False**

Briques logiques de **Panneau victoire**

Sensors: Panneau victoire | Add Sensor

Controllers: Panneau victoire | Add Controller

Actuators: Panneau victoire | Add Actuator

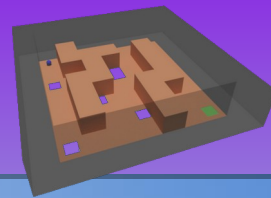
Logic 1: Toujours (Always) → And → Visibilité (Visibility) [Visible: False]

Logic 2: Message (Message) [Subject: victoire] → And (And.001) → Visibilité (Visibility.001) [Visible: True]

## 2 : Afficher le panneau en cas de victoire

- **Ajouter** un **Capteur Message** avec « victoire » comme sujet,
- **ajouter** un **Contrôleur And**,
- **ajouter** un **Actionneur Visibilité** avec **True**

# 10. Fermer le panneau de victoire par clic



## 3 : Désactiver le clic du panneau au départ

- Ajouter un **Capteur Toujours**,
- ajouter un **Contrôleur And**,
- ajouter un **Actionneur Edit Object** avec **Suspend Physics**

The screenshot shows the logic editor interface with three main sections: Sensors, Controllers, and Actuators. In the Sensors section, 'Toujours' (Always) is selected. In the Controllers section, 'And' is selected with 'And.001'. In the Actuators section, 'Edit Object' is selected with 'Dynamiques' and 'Suspend Physics' chosen. A 'Message' sensor is also visible with the subject 'victoire'. The 'Subject' field is highlighted with an orange box.

## 4 : Réactiver le clic du panneau en cas de victoire

- Ajouter un **Capteur Message** avec le sujet « victoire »,
- ajouter un **Contrôleur And**,
- ajouter un **Actionneur Edit Object** avec **Restore Physics**

Briques logiques de **Panneau victoire - Plan**

# 10. Fermer le panneau de victoire par clic



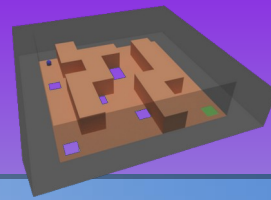
The screenshot shows the Construct 3 logic editor interface. It is divided into three main sections: Sensors, Controllers, and Actuators. Each section has a header with 'Sel', 'Act', 'Lier', and 'État' checkboxes. The 'Sensors' section contains two 'Mouse Eve' sensors: 'Mouse Over' and 'Left Button'. The 'Controllers' section contains an 'And' controller with 'State 1' selected. The 'Actuators' section contains an 'Edit Object' actuator with 'Restart' mode selected. Orange boxes highlight the 'Add Sensor', 'Add Controller', and 'Add Actuator' buttons, and the specific sensor, controller, and actuator settings.

5 : Redémarrer sur il y a un clic sur le panneau

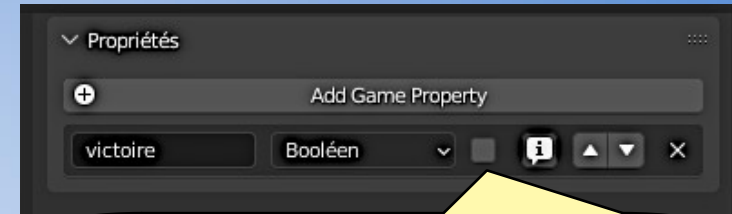
- Ajouter un **Capteur Souris** avec **Mouse Over**,
- ajouter un **Capteur Souris** avec **Left Button**,
- ajouter un **Contrôleur And**,
- ajouter un **Actionneur Scène** avec **Restart**

Briques logiques  
de **Panneau  
victoire - Plan**

# 10. Fermer le panneau de victoire par clic



Quand la bille passe l'arrivée, elle tombe et la détection de proximité de la bille avec le plateau génère le redémarrage de la partie sans que le panneau de victoire ai été cliqué. Le principe est d'interdire cette action en cas de victoire.



6 : Créer la **propriété victoire (binaire)** à la **Bille**

Vue Ajouter

Sensors  Sel  Act  Lier  État

Bille Add Sensor

Proximité Near

Skip 0 Niveau Tap Inverser

Propriété:

Distance 10.000 Reset Distance 11.000

Propriété Property

Evaluation Ty... Égal

Propriété: victoire

Valeur: True

Controllers  Sel  Act  Lier

Bille Add Controller

Expres... And

Controller visible at: State 1

NOT Near AND NOT Property

Actuateurs  Sel  Act  Lier  État

Bille Add Actuator

Scène Scene

Mode: Restart

Briques logiques de **Bille**

7 : Conditionner le redémarrage si victoire = False

- **Ajouter** un **Capteur Propriété** avec **victoire** sur **True**,
- **ajouter** un **Contrôle** Expression avec **NOT Near AND NOT Property**

# 10. Fermer le panneau de victoire par clic



Briques logiques de **Bille**

Sensors

Proximité Near

Distance 10.000 Reset Distance 11.000

Propriété Property

Message Message

Subject: victoire

Controllers

Expres\_ And

Controller visible at: State 1

NOT Near AND NOT Property

And And.001

Controller visible at: State 1

Actuators

Scène Scene

Mode: Restart

Propriété Property

Mode: Assign

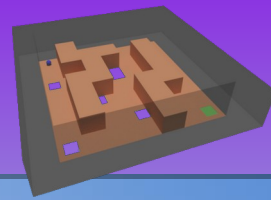
Propriété: victoire

Valeur: True

8 : Mettre la propriété **victoire** à **True** lors de la réception du message « **victoire** »

- **Ajouter** un **Capteur Message** avec le **sujet** « **victoire** »,
- **ajouter** un **Contrôleur And**,
- **ajouter** un **Actionneur Propriété** pour **assigner True** à **victoire**

# 11. Animer le panneau de victoire par des images-clés



Afin que le panneau de victoire apparaisse progressivement, nous allons l'animer avec deux images-clé (keyframe), une à la frame 0 et une à la frame 100.

1 : Passer sur le **bureau Animation**

2 : Sélectionner l'objet **Panneau victoire**

3 : Activer l'outil **déplacer** avec le **repère local**

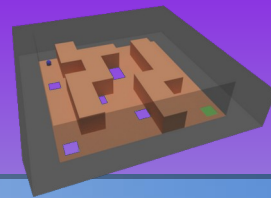
5 : Créer une image-clé  
**Insérer image-clé [I]**,  
cliquer sur **Position** puis  
refaire pour mémoriser la **Taille**

4 : Aller à la **frame 100**

The screenshot shows the Blender 2.80 interface with the following elements highlighted:

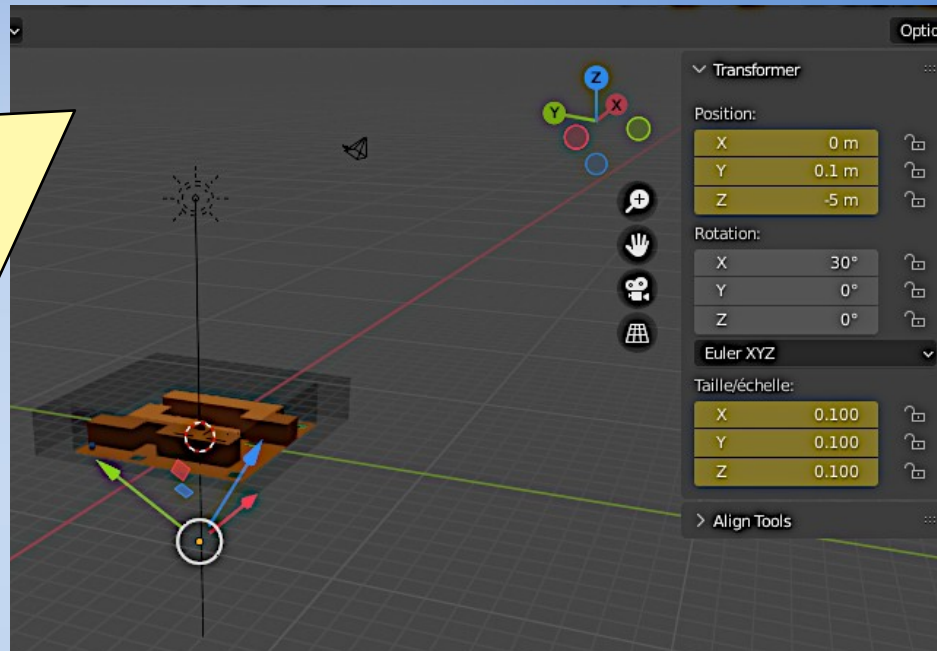
- Animation workspace:** The top bar shows 'Animation' and 'Scripting' tabs.
- Orientation:** The 'Local' orientation is selected in the top bar.
- Tool Shelf:** The 'Move' tool is active, and the local coordinate system (X, Y, Z axes) is visible.
- Outliner:** The 'Panneau victoire' object is selected in the Outliner panel.
- Properties Panel:** The Transform properties are visible, showing Position (X: 0 m, Y: -5.2125 m, Z: 9.3627 m) and Rotation (X: 30°, Y: 0°, Z: 0°).
- Timeline:** The timeline is set to frame 100.
- Insert Keyframe Menu:** The 'Insert Keyframe' menu is open, with 'Position' and 'Scale' selected.

# 11. Animer le panneau de victoire par des images-clés



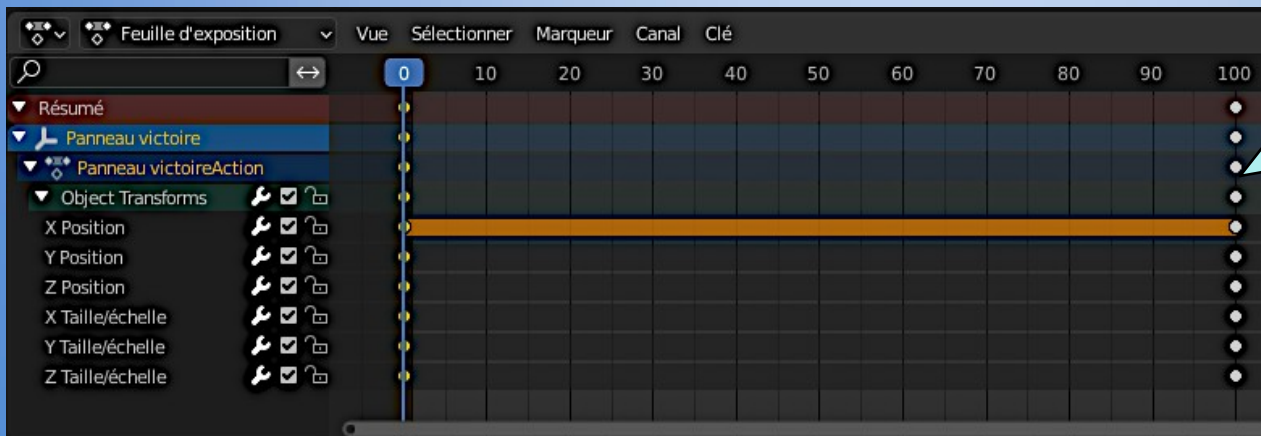
## 6 : Créer l'image-clé du départ de l'animation

- **Aller** à la frame **0**
- **changer l'échelle** de l'objet à **0.1**
- **déplacer** le plan avec le **trièdre local** pour le **caché derrière le plateau** (par exemple à la position  $(x,y,z) : 0, 0.1, -5$ )
- **insérer image-clé [I]** en **position**
- **insérer image-clé [I]** en **taille**



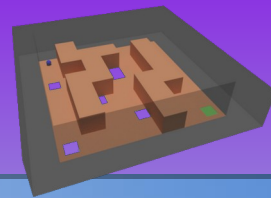
La **feuille d'exposition** indique les images-clé mémorisées.

Vous pouvez **jouer l'animation** avec **[Space]**.





# 11. Animer le panneau de victoire par des images-clés



Briques logiques de **Panneau victoire**

**7 : Ajouter l'animation lors de la victoire**  
**Ajouter** un **Actionneur Action** avec

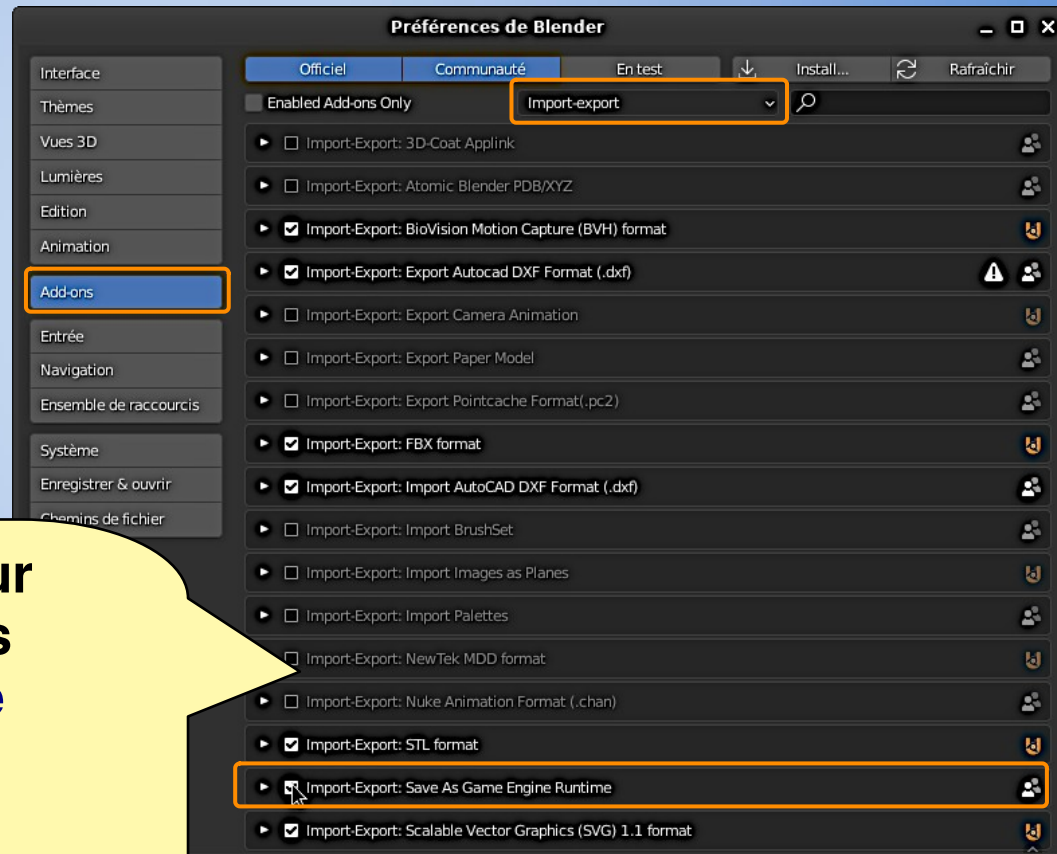
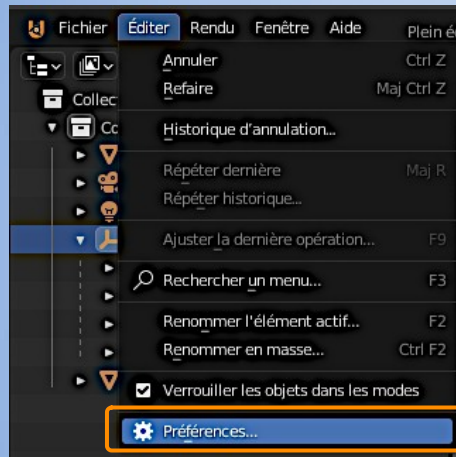
- Action : **Panneau victoire Action**
- Début : **frame 0**
- Fin : **frame 100**

**8 : Valider le comportement du panneau [P]**

# 12. Produire un exécutable



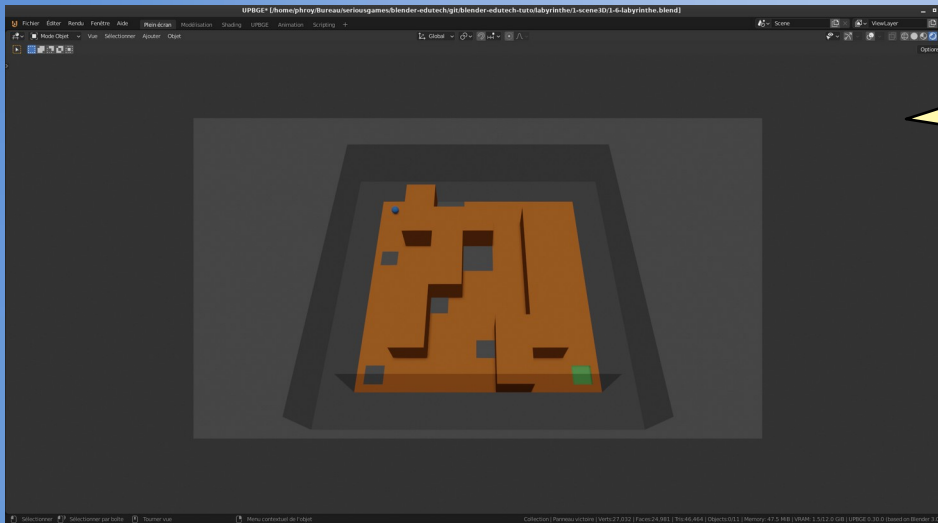
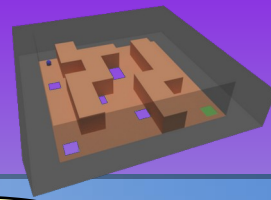
Afin de pouvoir exécuter le jeu sans Blender, il faut produire un exécutable. Le processus de compilation se fait sur chaque plateforme (GNU/Linux, Windows ou macOS) séparément. Ici c'est sous GNU/Linux, les différences entre les plateformes sont minimales.



## 1 : Installer l'extension pour produire des exécutables

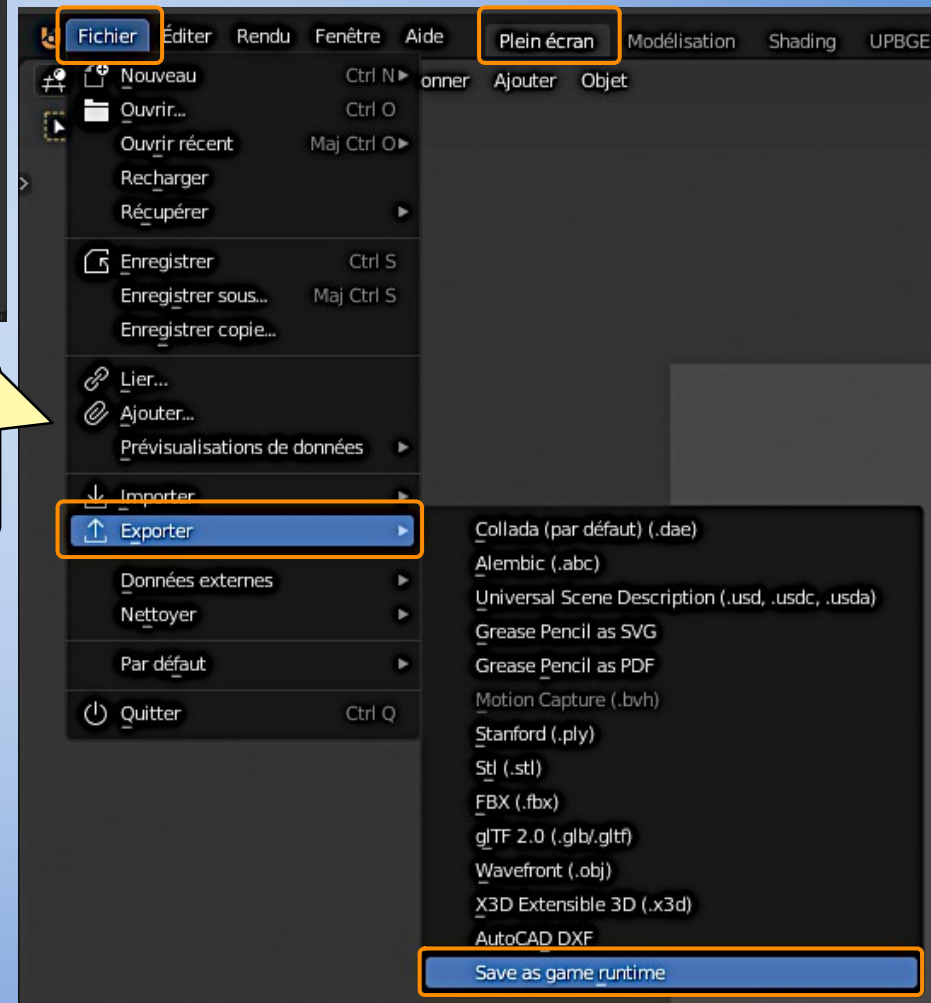
- **Menu Éditer** puis **Préférence**
- Onglet **Add-ons**
- Catégorie **Import-export**
- **cocher Save as Game Engine Runtime**

# 12. Produire un exécutable

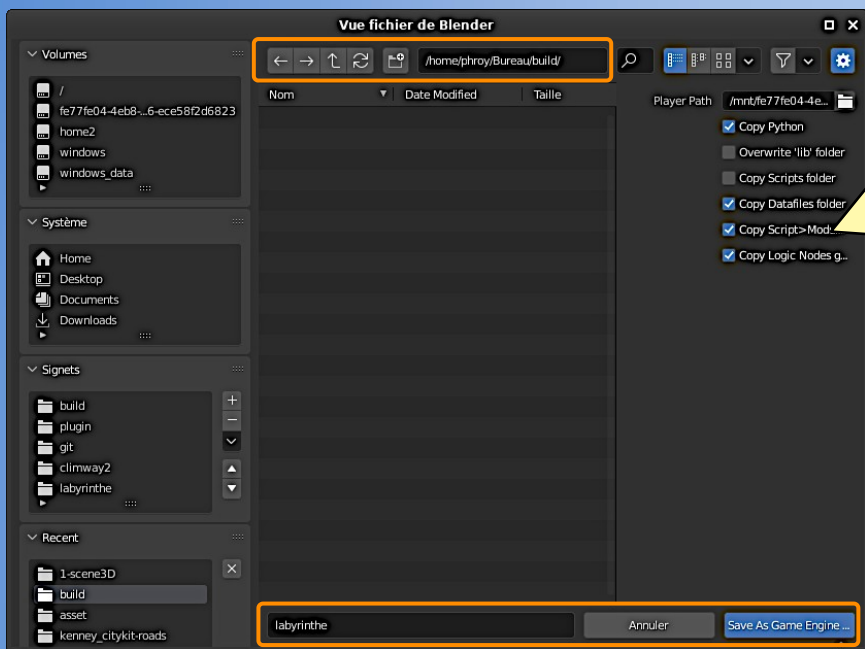


**2 : Passer sur le bureau Plein écran**

**3 : Exporter comme un exécutable**  
**Menu Fichier / Exporter /**  
**Save as game runtime**

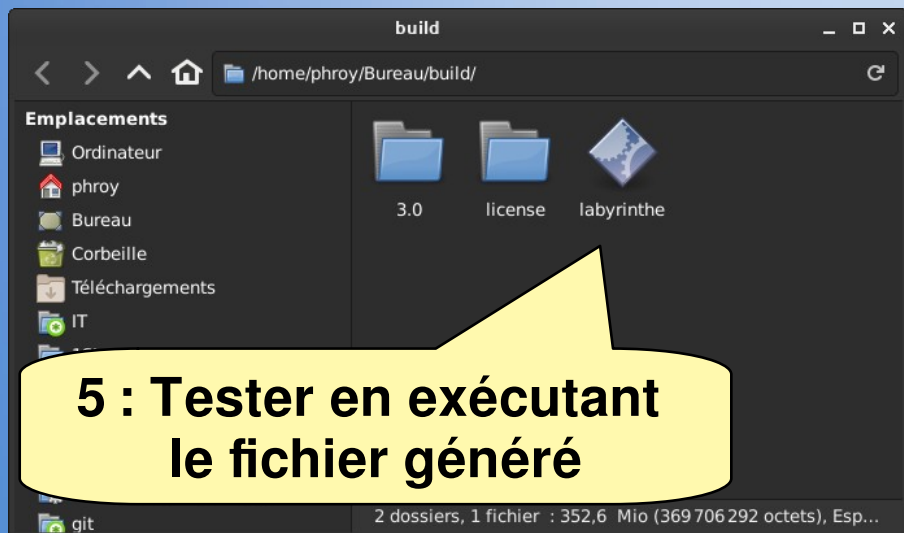


# 12. Produire un exécutable



**4 : Définir le répertoire contenant l'exécutable et ses fichiers ainsi que le nom de l'exécutable**

- Sous GNU/Linux, le nom de l'exécutable n'a pas d'extension.
- Sous Windows, le nom de l'exécutable a l'extension « **.exe** ».



**5 : Tester en exécutant le fichier généré**



**Bravo ! Vous êtes arrivé à l'issue de ce tutoriel.**