


# Casse brique game Maker

création d'un casse brique avec game maker

 Difficulté Facile

 Durée 3 heure(s)

 Catégories Jeux & Loisirs

 Coût 0 EUR (€)

## Sommaire

Introduction

Étape 1 - Création des sprites

Étape 2 - Création des objets : la raquette

Étape 3 - Création des objets : briques

Étape 4 - Création des objets : Bordure du terrain

Étape 5 - Création des objets : la balle

Étape 6 - Création de la room

Étape 7 - Etoffez votre création.

Notes et références

Commentaires

## Introduction

Voici un tuto pour coder un casse brique simple à l'aide de Game Maker

### Matériaux

### Outils

- Ordinateur
- Game Maker Studio

<https://drive.google.com/file/d/0B5tEZobBAb3zTFF5SjExMHQzUzA/view?usp=sharing>

## Étape 1 - Création des sprites

Tout d'abord il vas vous falloir créer des sprites pour vos éléments de jeux.

Pour ce casse-Brique nous auront besoin de 4 éléments (donc 4 sprites) :

- les briques
- la raquette
- la balle
- et les bordures du terrain pour éviter que la raquette ou la balle sorte du niveau par les cotés.

Bien penser à centrer l'origine de vos sprites et à appliquer un masque de collision le plus près possible de vos formes originale.

Les sprites sur le coté sont ceux qui ont été créer à la base pour les différents objets, mais vous pouvez en créer d'autre si vous le souhaitez.



## Étape 2 - Création des objets : la raquette

Comme pour les sprites, vous allez devoir créer un objet par élément de votre jeu. Nous allons d'abord créer chaque objet (raquette, mur, brique, balle), c'est important de créer chaque objet avant de commencer à les paramétrer en profondeur.

Une fois chaque objet créé nous allons nous intéresser à la raquette.

Tout d'abord dans l'événement create de votre raquette, initialisez quatre variables :

- nb\_brique à 0
- ismoving\_right à 0
- ismoving\_left à 0
- ismoving à 0

la première servira à savoir s'il reste des briques sur le terrain et les trois autres seront utiles pour le contrôle des mouvements de la raquette.

sur l'événement Step, entrez l'algorithme suivant : si nb\_brique = 0

```
{
  alors restart la room;
}
```

ensuite créer un événement "collision avec mur", et mettez comme le "bounce" avec les paramètres suivants : appliquer aux autres, précision précise, et rebondir sur tous les objets.

maintenant passons aux mouvements : pour tout ce qui concerne la gauche, créez un événement left pressed et left released. Dans left pressed, implantez l'algorithme suivant : si ismoving = 0

```
{
  start moving in a direction (droite, force -7, relative)
  set variable ismoving_left to 1
  set variable ismoving to 1
}
```

Ce bout de code signifie, que si la raquette ne bouge alors elle commence à se déplacer à gauche lorsque l'on appuie sur la flèche de gauche, il permet aussi d'empêcher certains bugs si jamais la flèche droite et la flèche gauche sont appuyées en même temps.

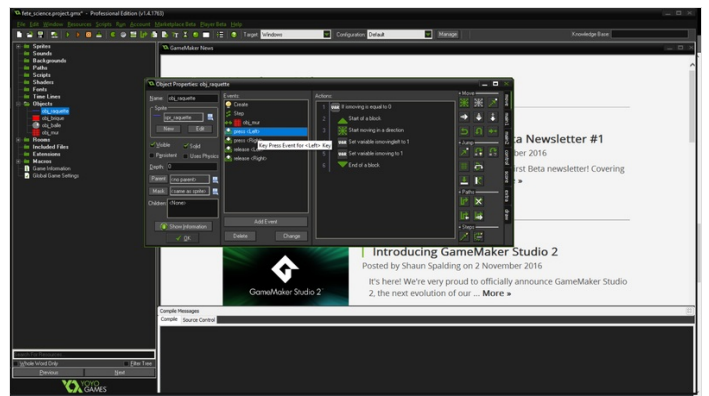
Pour left released le code ressemblera à ça : si ismoving\_left = 1

```
{
  start moving in a direction (droite, 7, relative)
  set variable ismoving_left to 0
  set variable ismoving to 0
}
```

Sachez que le 7 est une valeur arbitraire, vous pouvez la changer à tout moment pour moduler la vitesse de votre raquette. Par contre n'oubliez pas que la vitesse lors du pressage de touche doit être l'opposée de celle lors du release.

Pour faire le déplacement vers la droite il suffit d'inverser la puissance de déplacement et de remplacer tout les ismoving\_left, par des ismoving\_right.

Et voilà vous n'aurez plus à toucher la raquette.



---

## Étape 3 - Création des objets : briques

Ici nous irons plus vite car les briques sont simples à manipuler.

Tout d'abord, dans l'événement create de la brique il faut set la variable nb\_briques, à 1 relative, appliqué à l'obj\_raquette.

Ensuite, pour l'événement destroy, faire de même mais au lieu de 1 il faut set la variable à -1.

## Étape 4 - Création des objets : Bordure du terrain

Il n'y a rien à faire dans cet objet, ce sont les autres objets qui interagissent avec qui font toute les opérations.

## Étape 5 - Création des objets : la balle

Cet objet est le plus compliqué à faire mais aussi le plus important de votre jeu.

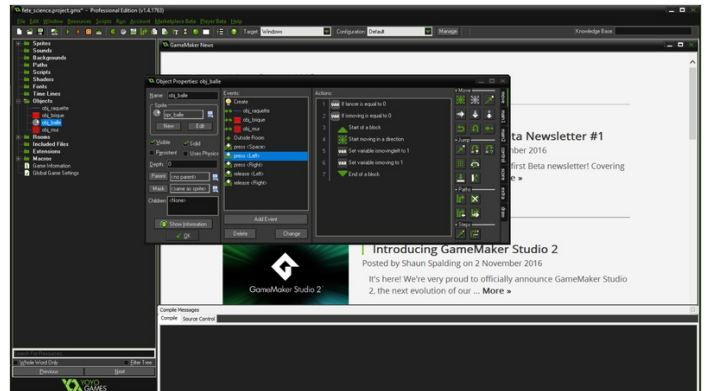
Tout d'abord, dans l'évènement create, il faut set des les variables ismoving, ismoving\_right, ismoving\_left, et une variable lancée à 0. Les variables ismoving serviront à manipuler la balle avant de la lancer.

Maintenant pour l'évènement de collision avec la raquette, il faut un bounce avec les paramètre suivants : appliquer à sois, bounce contre tout objet, et rebond précis.

Pour la collision avec une brique il faut un rebond comme pour la raquette, et ensuite détruire l'instance contre laquelle la balle entre en collision.

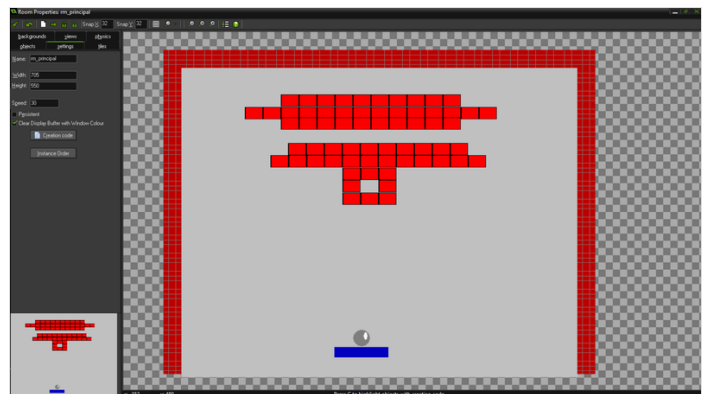
Pour la collision avec les bordure du terrain c'est plus compliqué. Si la balle n'est pas lancée alors il faut que le rebond soit appliqué sur la bordure et si la balle est lancée alors il faut l'appliquer à la balle

Si la balle sort des limites de la room, il faut restart la room  
Lorsqu'on appuie sur espace, si lancée vaut 0, il faut set la variable lancée à 1 et donner une vitesse verticale à la balle (dans mon jeu j'ai mis la vitesse de la balle à 5)  
enfin pour les actions sur les touches fléchée, c'est exactement le même algo que pour la raquette à ceci près qu'il faut vérifier si la balle est lancée, si lancée = 0 alors la balle peut bouger, sinon elle ne doit pas être influencée par les touches fléchée



## Étape 6 - Création de la room

Il ne vous reste plus qu'à créer une room qui vous servira d'espace de jeu, je recommande de mettre une vitesse de 30. Voilà la room que j'ai fait, mais vous êtes libre de faire un niveau comme vous le souhaitez



## Étape 7 - Etoffeiez votre création.

Maintenant que vous avez la base d'un casse brique, il est temps d'ajouter du son, des graphismes moins sommaires, des menus, un systèmes de scores etc... Bref amusez vous à rendre votre petit jeu un peu plus "pro"

## Notes et références

Les touches sont gauche droite et espace car cela permet de contrôler le jeu avec un Makey makey.