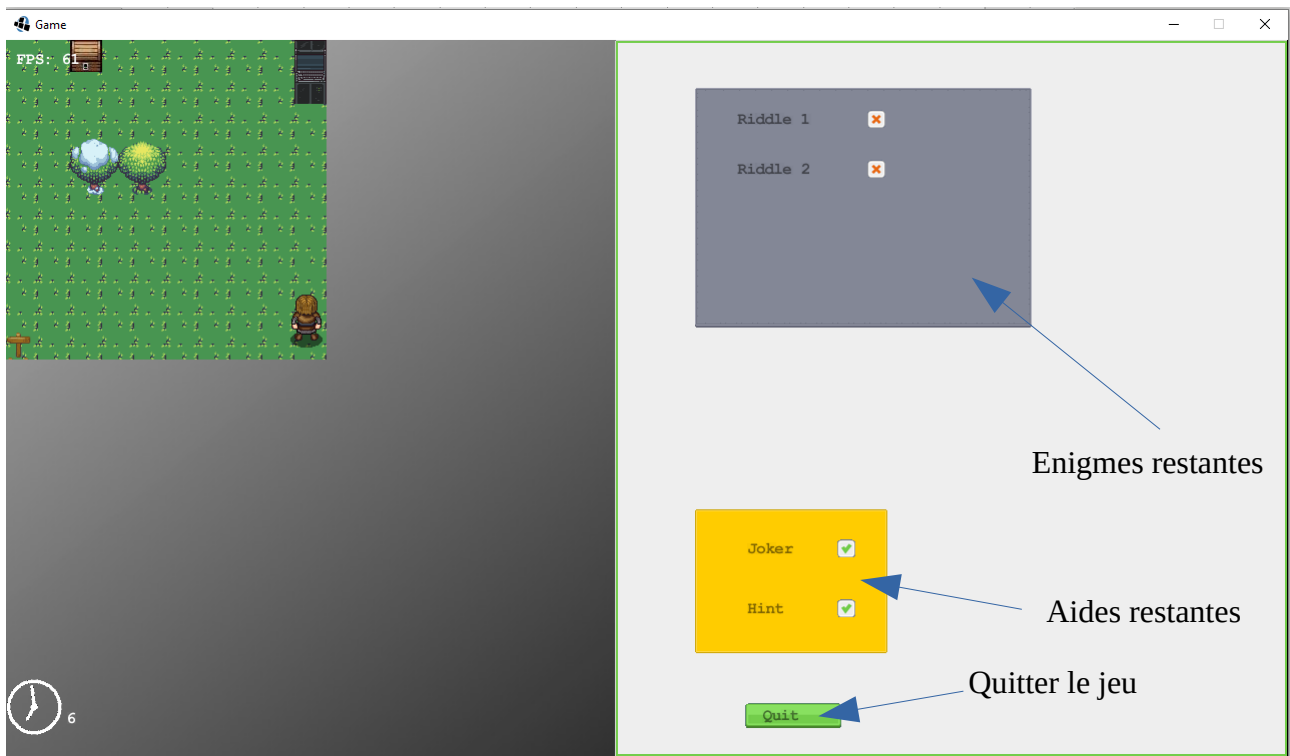


Documentation d'utilisation

Le jeu consiste à terminer un niveau remplis d'énigmes de code.



En haut à droite on peut voir les énigmes restantes, **X** nous informe que l'énigme n'est pas résolu.

En haut à gauche on peut voir les aides restantes, **V** informe que l'aide est toujours disponible.

Voici notre carte il faut examiner des éléments dans la carte pour lancer une énigme.

Dans le cas présent le coffre et le PC devront être examiné.

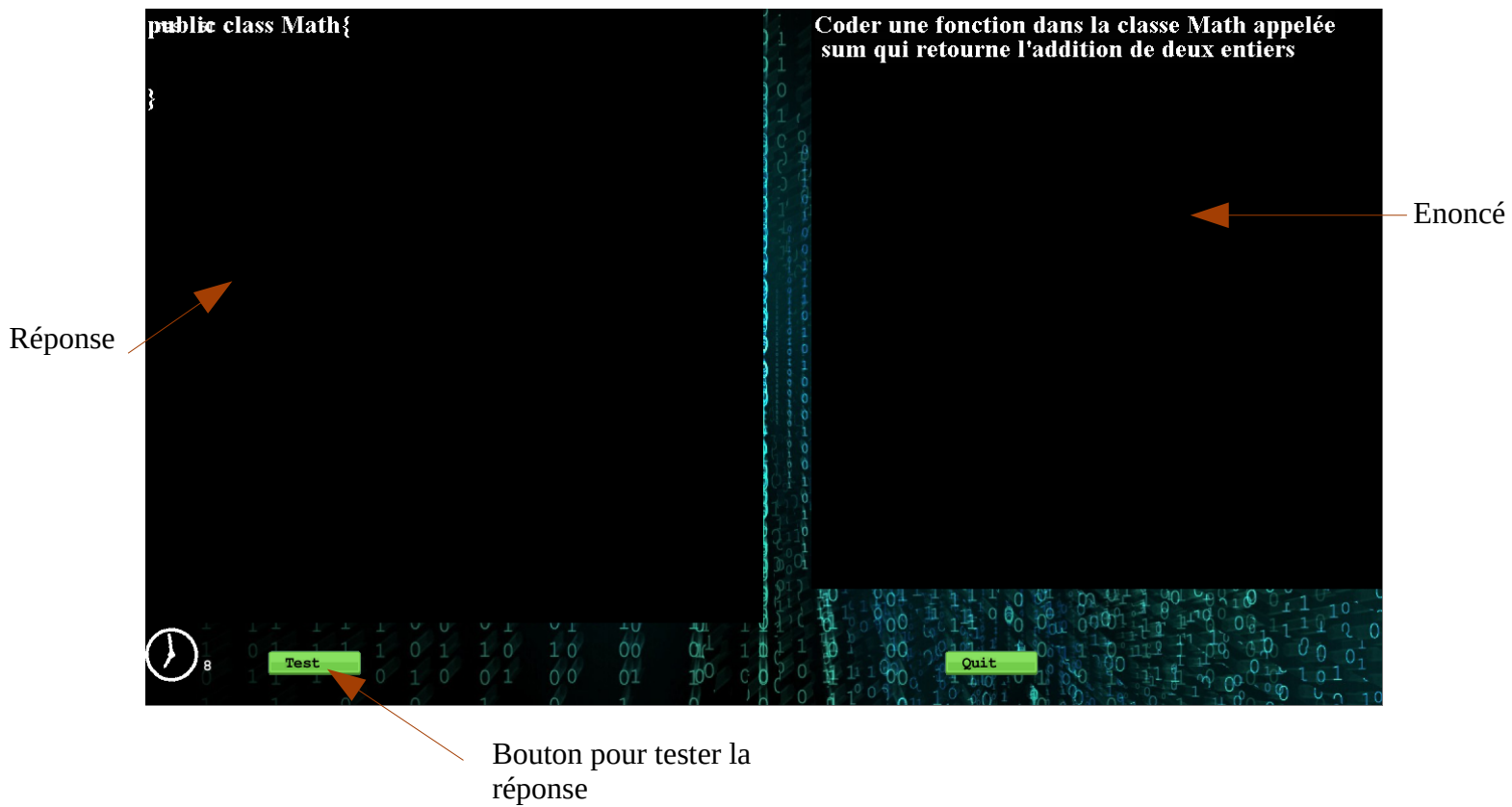
La touche espace permet d'examiner un élément une fois devant.

Le joueur peut se déplacer avec les flèches directionnelles sur le clavier.



Phase d'énigme

Après avoir examiner l'élément nous nous retrouvons dans une phase ou nous devons coder pour résoudre l'énigme.

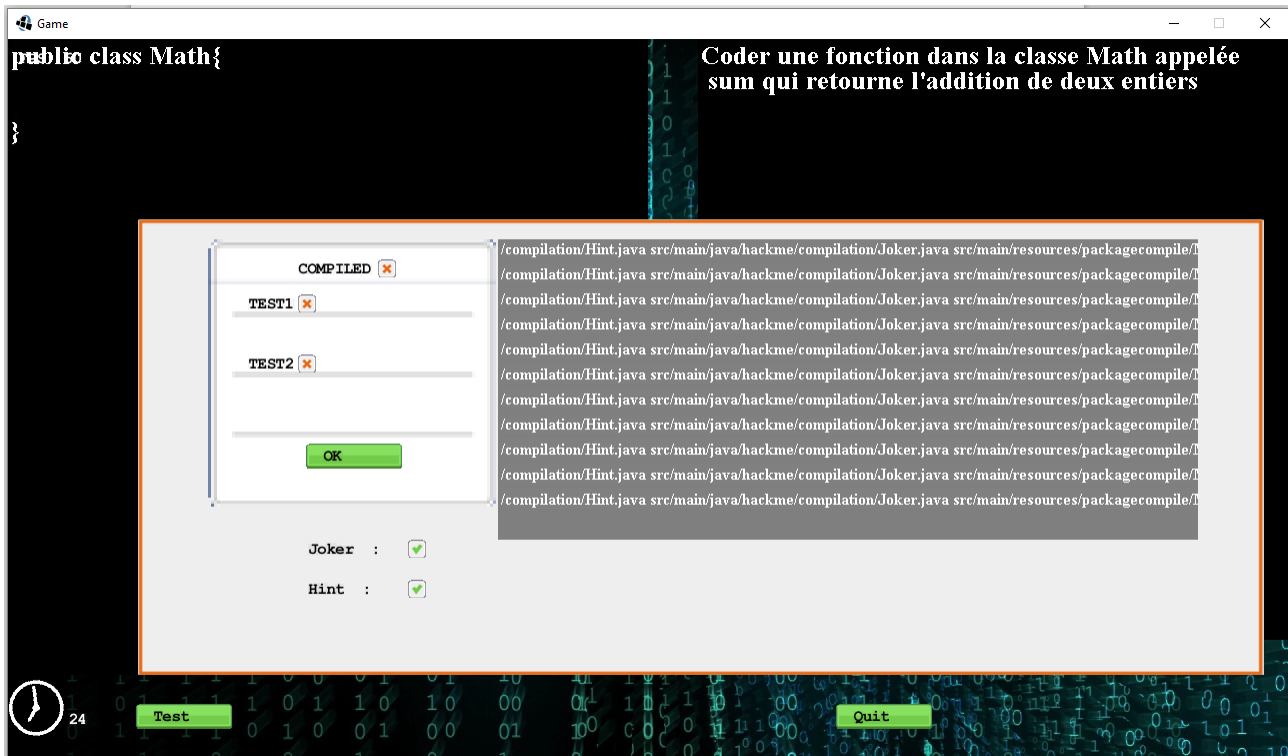


On peut naviguer dans les champs texte grâce au flèches directionnelles (Flèche du haut et du bas permettent de naviguer entre les lignes) .

Le bouton test permettra de compiler et de tester le code de l'utilisateur.

Le bouton Quit renverra l'utilisateur sur la carte.

Test

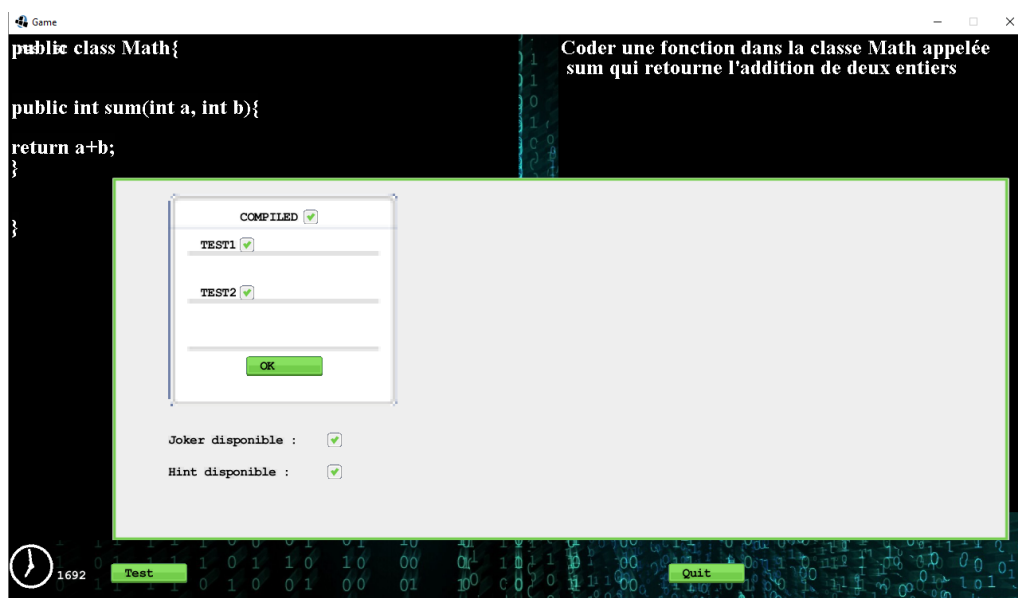


Après avoir lancer les tests des indicateurs s'afficheront pour prévenir si le code a réussi à être compilé et nous montrer quels tests sont passés.

Les aides restantes sont aussi affichées leurs utilisations seront expliqué dans une partie leur étant consacrée.

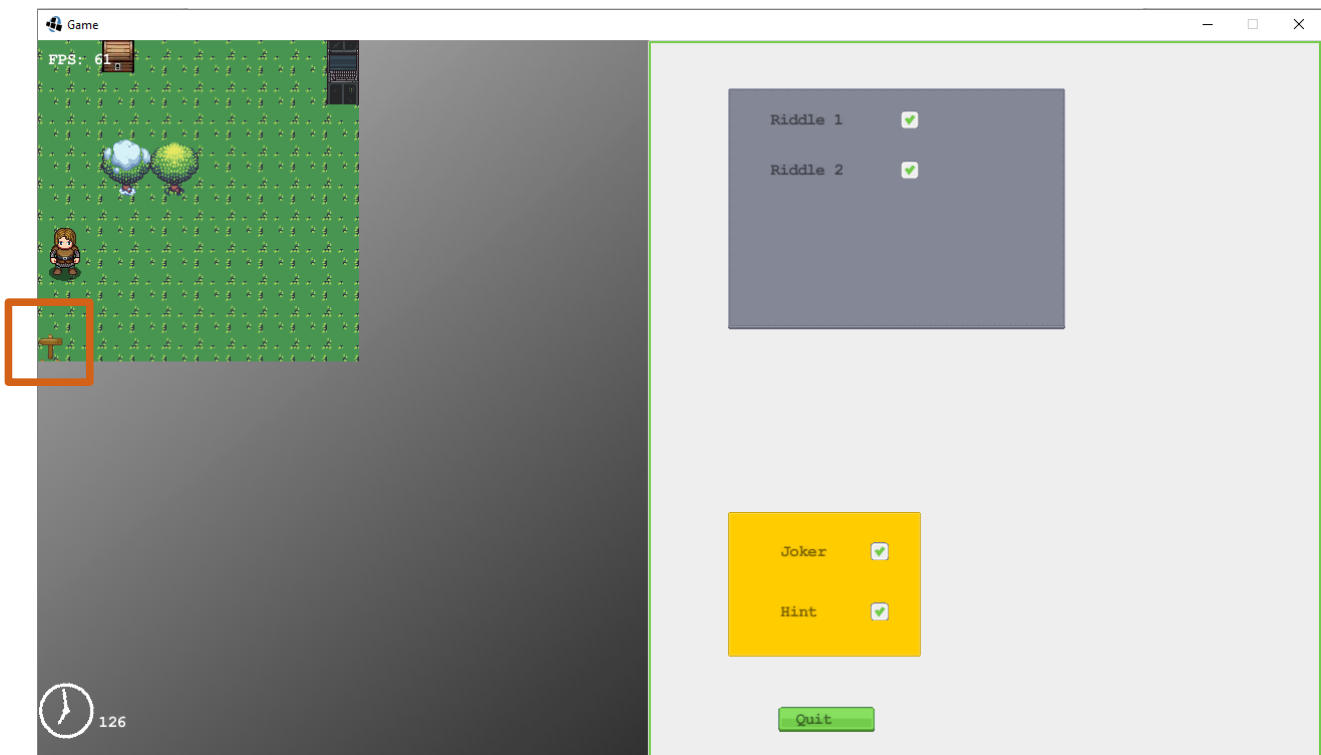
Si le code n'a pas passer la compilation un champ qui contient le stack d'erreur s'affichera , (il est conseillé de naviguer avec la flèche du haut et du bas pour naviguer dans le stack d'erreur) En appuyant sur le bouton «Ok» la fenêtre se fermera et l'utilisateur pour de nouveau coder.

Une fois les tests passés, en appuyant sur le bouton «Ok» l'utilisateur sera renvoyé sur la carte pour qu'il passe aux énigmes suivantes.



Après avoir finis les énigmes il faut se diriger vers la sortie pour que l'utilisateur termine la partie.

Sur notre carte la sortie est désigné par le petit panneau en bas.



Après avoir atteint la sortie l'écran final apparaît il faudra appuyer sur espace pour fermer le jeu. Votre score sera sauvegardé et envoyé en ligne. (Le score correspond au temps)



Aide

Il existe 2 aide sur le jeu, le Joker et le Hint.

Et chacune de ces aides ne pourront être utilisées qu'une seule fois pendant toute la session de jeu.

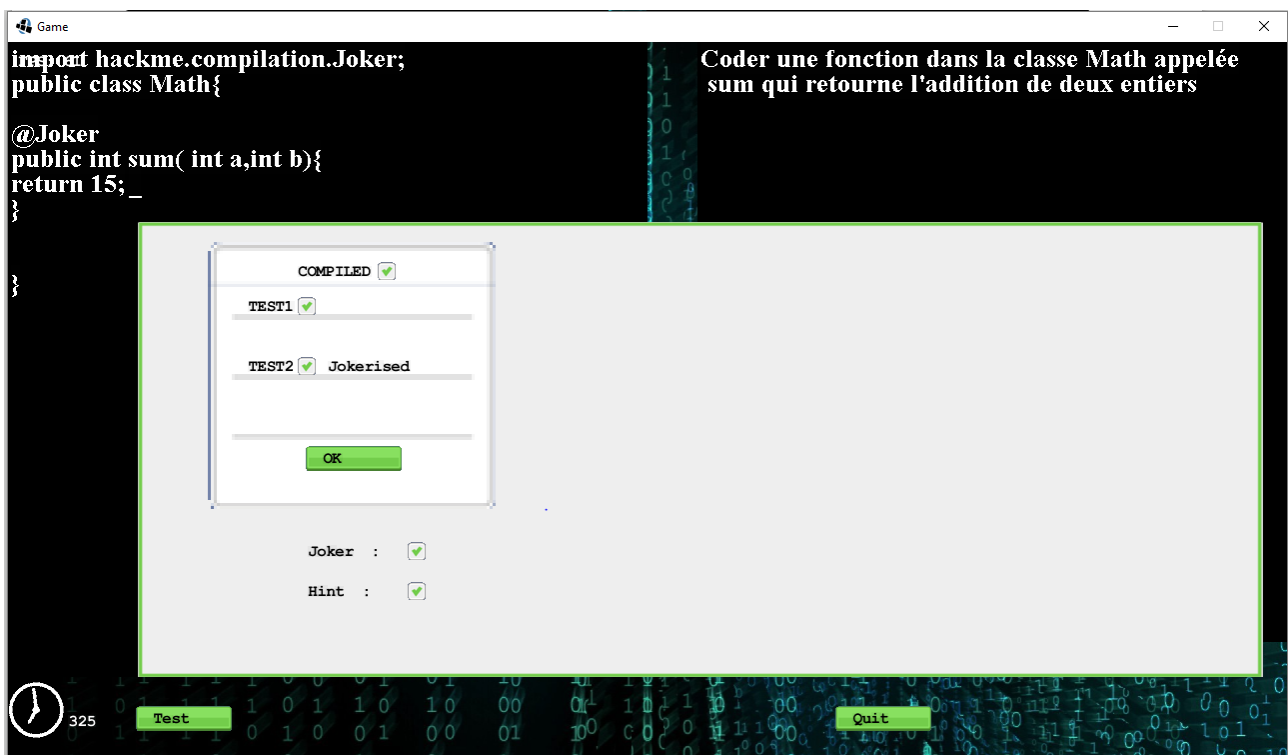
Joker

Le Joker permet d'ignorer le premier test qui ne passe pas et permet a l'utilisateur de ne pas réussir un test et de quand même réussir une énigme.

Pour l'utiliser il faut mettre l'annotation `@Joker` au dessus de la fonction codée.

Le test passé sera noté comme «Jokerised».

Important: Il ne faut pas oublié d'importer l'annotation.

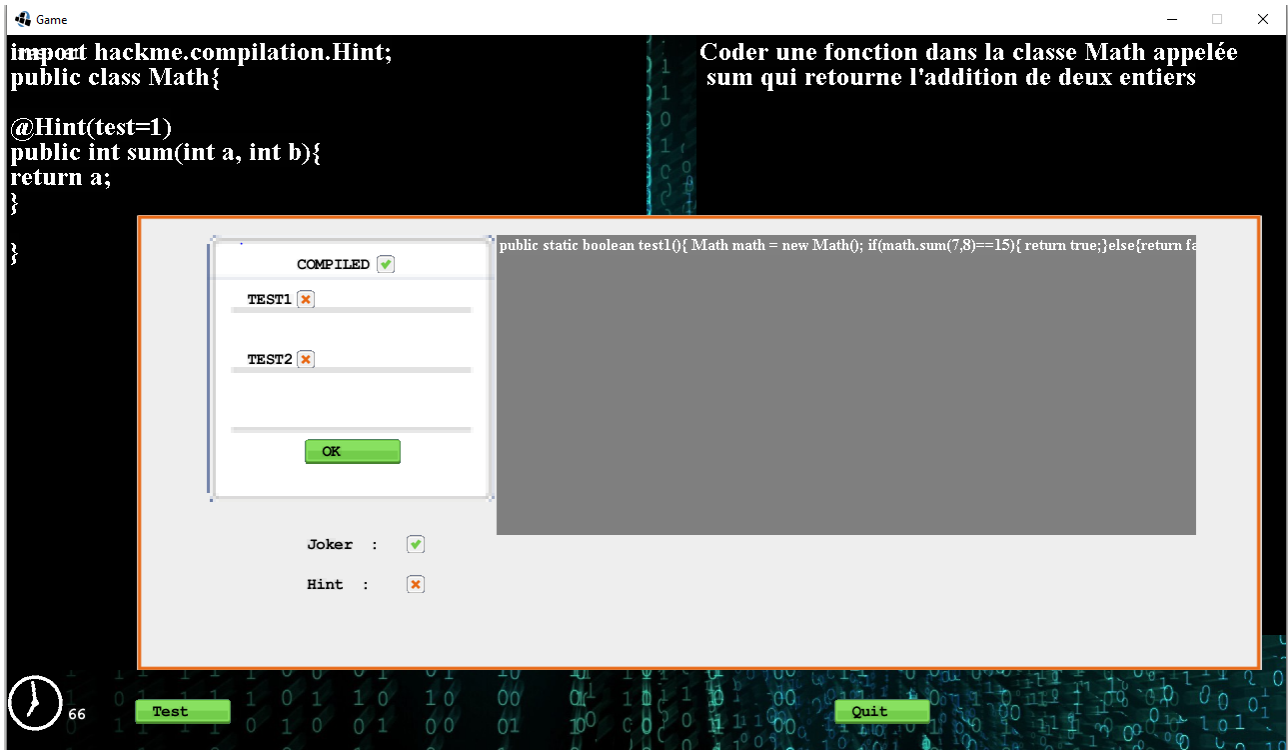


Hint

L'aide Hint permet d'afficher comment est fait le testX, il suffit de d'écrire @Hint(test = X), X représentant le numéro du test.

Vous pourrez désormais voir comment votre fonction est tester.

Important: Il ne faut pas oublié d'importer l'annotation.



Création d'une carte

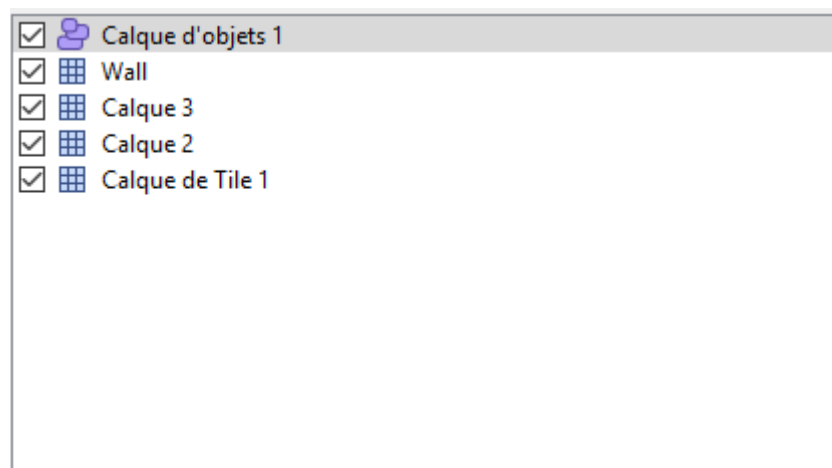
Pour créer une carte nous utilisons le logiciel Tiled V0,92 .

Les cartes devront être créées sous un certain format:

Base64 (Compressé par gzip)

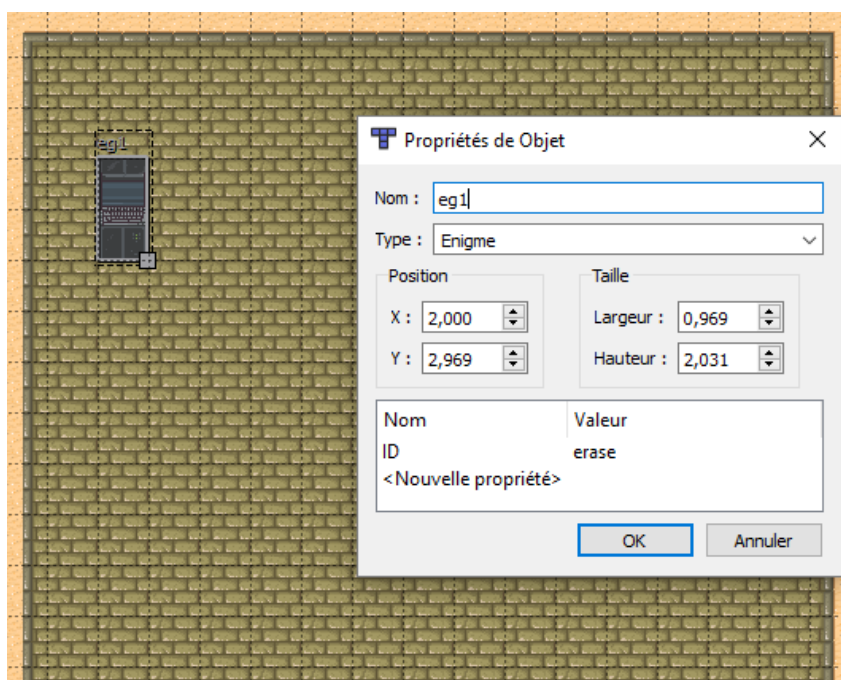
Les calques devront être rangés sous cette forme.

L'avant dernier Calque aura pour obligation de s'appeler Wall



Le calque d'objets devra contenir des objets de Type Enigme et Sortie

Les objets de type Enigme devront posséder une propriété ID avec le nom du JSON qui correspond à l'énigme.



Et les Sprites utilisés par la map Tiled devront être ranger dans un dossier

Ce qui nous donne se format.

ge	Affichage	Gestion	
PC > Bureau > Projet > Projet_annuel_3AL1_2018 > Maps > map			
Nom	Modifié le	Type	Taille
Enigme	22/07/2018 12:35	Dossier de fichiers	
Tile	22/07/2018 12:45	Dossier de fichiers	
map.tmx	21/07/2018 23:44	Tiled.tmx	3 Ko

Les Sprites sont rangé dans Tile

La Map devra obligatoirement s'appeler map.tmx

Tous les JSON d'énigme devront être upload sur l'application web

Le dossier contenant les Sprites de la map devront être zipper(.zip obligatoirement) et upload sur l'application web.

La map.tmx devra être upload aussi sur l'application web.

JSON Enigme

```
{
  "_function_name" : "sum",
  "_enonce" : "Coder une fonction sum dans la classe math qui prend deux int et qui les additionnes",
  "_nameTest1": "test1",
  "_nameTest2": "test2",
  "_nameTest3": "",
  "_blocTest1": "public static boolean test1() { Math math = new Math(); if(math.sum(7,8)==15) { return true;}else{return false ;}}",
  "_blocTest2": "public static boolean test2() { Math math = new Math(); if(math.sum(17,8)==25) { return true;}else{return false ;}}",
  "_blocTest3": "",
  "_className": "Math"
}
```

function_name : contient le nom de la fonction a codé

_enonce : l'enonce pour l'utilisateur

_nameTestX contient le nom de votre fonction de test

_blocTestX contient la declaration de votre fonction de test (Toujours public static boolean!)

_className contient le nom de la classe que l'utilisateur devra créer et coder la fonction demandé a l'interieur