








7.5

Révision de code du fichier **render-service-6.ts** par **David Tremblay (équipe 5)**

Ligne(s)	Défaut(s) trouvé(s)
Fichier entier	Le service est BEAUCOUP trop long avec 685 lignes. Il est évident que le nom est erroné et que ce fichier se charge de presque l'intégralité du jeu de curling. Plusieurs autres services devraient être créés pour se charger de la plupart des méthodes (voir prochains points).
Fichier entier	Toutes les méthodes "handles" devraient être dans un service game-manager.service et non dans le service de render. Toutes les méthodes calculant des angles ou autres paramètres devraient être aussi sorties du render. Le render ne devrait servir qu'à render la scène.
Fichier entier	Un service s'occupant des différentes caméras devrait être créé puisque ce n'est pas au render-service de s'en occuper.
Fichier entier	Certaines fonctions indiquent le type explicite de retour même en cas de void tandis que d'autres n'indiquent pas le type de retour. Il manque de cohérence et la première méthode est recommandée.
Fichier entier	Tous les strings magiques ou nombres magiques devraient se trouver dans un fichier de constantes dans un dossier commun du jeu (la plupart des nombres dans les attributs globaux par exemple).
Fichier entier	Il serait trop long de nommer chaque ligne mais TS LINT indique presque 60 erreurs de missing whitespace dans le document. Par exemple, un whitespace est manquant à la fin de la ligne 93 juste après les paramètres du constructeur
Fichier entier	Les else devraient être sur la même ligne que la parenthèse bouclée fermante du if correspondant. 
Fichier entier	Il y a environ 30 let qui devraient plutôt être des const puisqu'ils ne sont jamais réutilisés. Par exemple, les lignes 319 et 320 devraient être des const (trop long de tous les nommés). Il y a en fait que 2 seuls mots clés const dans l'entièreté du service de presque 700 lignes et ceci devraient sonner l'alarme.
Fichier entier	La moitié du document utilise " " tandis que l'autre utilise ' '. La dernière méthode devrait être celle utilisée pour tout le document (ligne 151 et 488 par exemple). Manque de cohérence
3	Ce service est mal nommé. J'ignore ce qu'il fait en ne regardant que son nom et même en regardant ou il est utilisé dans le fichier.
14	Chaque fois qu'on veut retirer un objet de la scène, nous appelons cet attribut public qui ne fait que déplacer l'objet très loin en dehors de la scène. Ceci est une très mauvaise façon de procéder et on devrait plutôt retirer l'objet définitivement de la scène en enlevant le child correspondant de celle-ci.
17	On définit une valeur approximative de PI en attribut global alors qu'on devrait utiliser Math.PI tout simplement.
22	Cet attribut privé devrait être avec les autres de ce type (après la ligne 42).
37	Ceci est le seul attribut déclaré public. Malgré que ceci soit la bonne chose à faire, les attributs devraient tous être déclarés publics ou non pour avoir une cohérence dans le code.
40	L'indentation de l'attribut est incohérente avec le reste.
43	L'attribut throwTest devrait être renommé isThrowing
44	L'attribut est mal nommé et n'est aucunement un test mais plutôt un simple booléen. On devrait enlever cet attribut et plutôt regarder quel type de caméra est utilisé pour décider la condition du if/else à réaliser dans handleKeyDown. L'attribut devrait être renommé isPerspectiveCam
8 à 40	Les attributs devraient être explicitement déclarés publics.
27, 28	Ces couleurs n'ont aucune raison d'être déclarées en variables globales.  devraient seulement être dans un fichier de constantes. De plus, on peut utiliser 'white' et 'black' comme couleurs pour faciliter la compréhension du code.
62	Ceci est un des deux seuls attributs utilisant la notation avec un underscore. Mauvaise cohérence du code. De plus, _wf n'est pas un nom d'attribut explicite. En regardant le code, on constate que le booléen réfère à un wireframe mais on ne sait toujours pas ce que ça fait.
82	Bien que cette option soit aussi acceptable, il aurait été intéressant d'utiliser les types déjà existants de THREE.js pour les effets sonores puisque c'est un jeu majoritairement en THREE.JS ( <a href="https://threejs.org/docs/#api/audio/Audio">https://threejs.org/docs/#api/audio/Audio</a> ).
87	Il faudrait renommer allThrown pour respecter la cohérence du code.

8 à 90	Ce service contient un nombre tout à fait aberrant d'attributs <del>globaux</del> étant donné un maximum d'environ 6 idéalement et il y en a presque 70. Tous les attributs qui sont utilisés qu'une seule fois ou qui ne change pas n'ont rien à faire en tant qu'attribut global (par exemple PI, cameraX, colorWhite, stoneRadius, etc.). Il serait trop long de tous les nommer, mais presque tous peuvent être facilement retirés avec une restructuration de code et une librairie de constantes.
8 à 90	Quelques-uns de ces attributs ne sont pas instanciés au départ alors qu'il le pourrait pour éviter du codage inutile. Par exemple, l'attribut  _allThrown devrait être instancié à false dès le départ et non dans le init.
33, 107	Les commentaires sont sur le côté au lieu d'en haut comme tous les autres. Il semblerait qu'il y a aussi d'autres pourcentages mais ceux-ci ne sont pas identifiés. Mauvaise cohérence 
158	On devrait éviter d'utiliser des event listeneners. Il faudrait utiliser les events d'angular idéalement.
111 à 113	Nous avons trois attributs publics pour la position de la caméra ainsi que trois attributs privés  et ils sont égaux. Pourquoi ne pas juste utiliser les attributs publics ou privés?
122 à 157	Certains commentaires d'une ligne utilisent /* *\ alors que d'autres utilisent //. De plus, les commentaires sont non pertinents (ligne 139 et 140 par exemple).
150, 151	Il faudrait éviter les getElementById ou getElementByTag ainsi que la manipulation du DOM. Il faudrait plutôt utiliser viewChild ou viewChildren.
221 à 235	En regardant le code, on constate qu'il ne semble n'y avoir aucun autre type de stone. La méthode devrait donc être renommée loadCurlingStones ou loadStones. De plus, la méthode est de type void mais contient un return qui ne retourne rien et qui devrait être retiré.
459	Nous avons une méthode pauseMusic() ne contenant qu'une seule ligne et qui n'est appelée qu'une seule fois. Il semble donc plus optimal de tout simplement pauser la musique à l'endroit où la méthode est appelée.
229, 244	Il faudrait renommer le paramètre wireFrame pour respecter la convention. 
282 à 295	On remarque que presque tout le code est dupliqué. Il est dur de se prononcer sans voir l'intégralité des fichiers du projet mais il semble possible de seulement alterner la visibilité de la barre et appeler la méthode resetStoneSpeed() lorsque qu'on alterne de false vers true et ainsi ne faire qu'une seule méthode.
381 à 403	Pourquoi y a-t-il un test  dans le render-service.ts ? Ce test devrait être dans un fichier test render-service.spec.ts.
381 à 403	Aussi, la déclaration du booléen collided est inutile. Il semble plus approprié de simplement retourner true si on entre dans le if et sinon else afin d'éviter d'avoir des lignes de codes superflus.
428 à 434	On peut mettre les noms des lumières et leurs vecteurs associés dans un tableau se trouvant dans le fichier de constantes et ainsi utiliser une boucle for pour réaliser cette méthode évitant ainsi de la duplication de code.
443 à 449	Même principe que précédemment. Nous pouvons utiliser le même tableau de constantes de noms de lumières et utiliser une boucle for.
58, 68, 80, 89, 598	Nous avons un attribut useAngle initialisé puis mis à false à la ligne 102 et jamais réutilisé. Nous avons aussi les attributs textureLoader, mouseY et created qui ne sont jamais réutilisés. On pourrait penser que certains de ces attributs sont utilisés dans d'autres fichiers mais la plupart sont déclarés privés et n'ont aucun get ou set pour y accéder. Code mort
451 à 457	Tous les effets sonores devraient être dans un service déduit à cet effet et non dans le service du render.
463 à 473	Le return dans le cas du else ne sert à rien et devrait être retiré.
484 à 493	Les changements de styles devraient se faire dans un fichier css et non dans le render.
636, 637	Ce code devrait être retiré. Code mort
163, 189, 200, 206, 212, 326	Toutes ces méthodes devraient être déclarées public, private ou protected.
34, 108, 318, 636, 637	Les commentaires doivent commencer avec un espace.