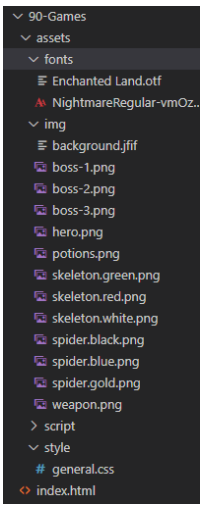
DungeonMaster

Partie 3 :

Dans cette dernière partie, nous allons utiliser l'ensemble de notre travail afin de réaliser un jeu fonctionnel. Le template du projet est fourni avec l'ensemble des éléments dont vous pourriez avoir besoin qui sont identifiés grâce à des **id**. vous devez utiliser uniquement le javascript natif avec vos connaissances en POO et en écriture modulaire. Aucune librairie ou Framework n'est tolérée.

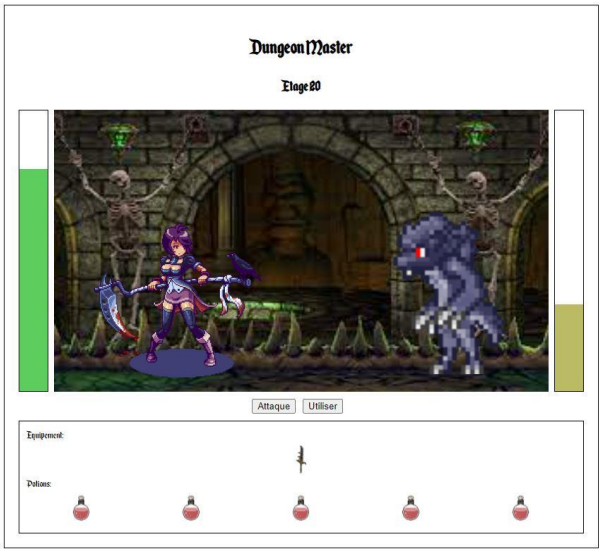
Le travail demandé est suffisamment conséquent pour que chaque membre du groupe soit en mesure d'expérimenter chacun des problématiques dont nous avons discuté en cours. N'oubliez pas de décomposer au maximum les difficultés que vous rencontrez et n'essayez pas de tout faire en même temps !



L'arborescence du projet est fournie et vous propose déjà un fichier de template (**index.html**), une modification du style (**general.css**) et des Sprite (**images**) permettant de figurer les différents ennemis et objets dont vous pourriez avoir besoin.

Vous êtes totalement libre de modifier le squelette mis à votre disposition et il en sera tenu compte dans la correction.

Pour ce projet, vous devrez prévoir l'affichage de 3 types d'ennemis différents dont nous avons créés les classes lors de la partie précédente.



Au niveau de l'interface :

- L'étage est mis à jour à chaque passage au niveau suivant
- L'écran principal contiendra votre héros à gauche et l'ennemi qui sera mis à jour à chaque niveau à droite
- De chaque côté de l'écran principal, figure la barre des points de vie
    - A gauche de notre héroïne
    - A droite de la créature à abattre
- En dessous, figurera des boutons vous permettant d'effectuer les différentes actions de votre personnage :
    - Attaquer : pour taper sur l'ennemi – visible uniquement quand il y a un ennemi
    - Utiliser : pour utiliser une potion rendant des points de vie – visible uniquement quand vous possédez des potions
    - Suivant : qui permet de passer la salle suivante – visible uniquement quand l'ennemi est mort
- Un emplacement pour votre inventaire
    - La première partie contiendra les armes que vous obtiendrez sur les monstres (4 au max)
    - La seconde contiendra les potions que vous obtiendrez aussi sur les créatures (10 max)
- Enfin, si vous perdez un écran apparait (**#lose** dans le CSS) afin de rechargez la page et de recommencer une partie

Le principe de fonctionnement :

- Lors du démarrage d'une partie :
    - Je pense à indiquer l'étage
    - J'initialise mon personnage
    - Selon l'étage j'affiche dans une couleur aléatoire le monstre correspondant
    - J'active les évènements sur les boutons qui doivent être visible

- A chaque clique sur le bouton attaquer :
    - Je récupère l'attaque du personnage et je la retranche au point de vie du monstre – sa défense
    - J'enchaine directement sur la riposte du monstre en réalisant la même action à l'envers
    - Je permets l'utilisation de potions afin de récupérer des points de vie lors du clique sur le bouton « utiliser »

- Quand le monstre meurt :
    - De manière aléatoire je génère un taux de récupération d'une potion ou d'une arme. Le taux étant différents selon le type de monstre
    - Je rends visible le bouton permettant de passage à l'étage suivant

- Au 10ème étage (5ème étage lieutenant) !
    - J'affiche la créature correspondante
    - Ces créatures possèdent des chances de récupération plus élevé que les monstres bases

- Après chaque boss de palier (étage multiple de 10) :
    - J'applique un coefficient afin de rendre l'attaque des prochains monstres et leur nombre de point de vie plus importants

Création des fonctions du jeu ou d'une classe les regroupant :

Ce point vous permettra de développer les fonctions de jeu pour réaliser une partie complète jusqu'au décès de votre personnage.

Si vous manquez de temps ou de pratique, l'ensemble de ces informations pourra donner lieu à un affichage console qui vous donnera moins de points qu'un affichage dans le template fourni. N'oubliez pas de revenir sur les classes déjà créées afin de permettre leur affichage dans le DOM du template fourni !!!!

L'objectif de cet examen est la réalisation de classes cohérent et utilisable dans ce dernier point. Je noterai moins sévèrement cette 3ème partie qui fera l'objet d'un coefficient moins important mais qui peut vous permettre de rattraper des catastrophes sur les évaluations ou autres TP notés.

Si vous prenez la décision de ne pas créer une classe de jeu, je vous invite à utiliser l'écriture modulaire en exportant l'ensemble des fonctions que vous allez créer ici afin de bien séparer les déclarations de leurs utilisations.

Dans ma correction, j'ai construit 4 fonctions :

- La première prend en paramètre le container et le titre contenant l'étage et me permet d'initialiser une partie et de retourner dans un objet tous les éléments dont j'ai besoin dans mes classes à savoir :
    - L'étage
    - Le héro
    - Le monstre
    - Les 3 boutons (attaque, utiliser et suivant)
    - Les sacs d'équipement (potions et armes)
    - Le container (où j'ajoute mes monstres)
    - Les points de vie de base de chaque créature et de mon héros (afin de calculer le pourcentage de vie perdu)

- La seconde prend en paramètre l'objet Game que je viens d'initialiser au-dessus et permet la gestion de l'affichage et de l'obtention des récompenses lorsqu'un monstre meurt

- La troisième prend en paramètre l'objet Game et le type (monstre ou héro) afin de mettre à jour l'affichage de la barre de vie

- La dernière prend en paramètre l'objet Game et me permet de programmer ce qu'il se passe quand je change d'étage :

    - Augmentation du numéro d'étage ou de salle
    - L'affichage du bouton attaquer (disparu lorsque le montre meurt)
    - L'augmentation des points de vie grâce à un coefficient d'augmentation défini par calcul (Etage / 10)
    - Quel monstre je dois afficher et où en fonction de l'étage
    - Remise à 0 des points de vie du monstre

Enfin, j'ai créé un fichier de jeu que je vais utiliser pour définir les réactions aux actions que l'utilisateur va effectuer.

Je vais aussi initialiser mon super objet Game qui regroupera l'ensemble des informations nécessaires au bon fonctionnement de mon jeu.

Dans ce script, j'ai géré ce qu'il se passe quand :

- Je charge la page : initialisation de la partie
- Je clique sur le bouton attaquer : gestion des dégâts, de la mort du monstre ou de la mort du héros
- Je clique sur le bouton utilisation : gestion des points de vie de l'utilisateur (maximum 100 et barre de vie à modifier) et de son sac (suppression d'une potion de mon héro affichage compris)
- Je clique sur le bouton suivant : j'appelle ma fonction calculant et préparant le nouvel étage

Barème

Consignes	Notes
Création des fonctions nécessaires à la réalisation d'une partie	/20
Partie fonctionnelle	/5
Prise en compte de la partie au niveau de l'affichage	/5
Modification des classes pour prise en compte du DOM	/7
Lisibilité / Structure du code / Arborescence	/5
Effort globale	/3

Le détail que je vous ai fourni pour la construction des classes ou des fonctions peut varier selon les solutions que vous allez trouver. Certaines propriétés peuvent apparaitre pour faciliter la gestion du DOM dans vos classes par exemple. Si c'est le cas, et qu'une telle propriété n'est utilisé qu'au sein de la classe, vous n'êtes pas obligé de créer les accesseurs et mutateurs correspondant. Attention les variables statiques n'ont pas de getters / setters.