

Projet Final: « *Roguelike* »

Introduction

Le genre « Roguelike » est un type de jeu dans lequel « le joueur explore un donjon infesté de monstres qu'il doit combattre pour gagner de l'expérience et des trésors » [[Wikipedia](#)]. Lorsque le joueur échoue, il perd sa progression et doit démarrer du début.

À l'instar de plusieurs jeux populaires tel que *The Binding of Isaac*, *DCSS*, *Hades*, *Faster Than Light*, *Slay the Spire* et même *Diablo*, vous devrez créer un jeu de style « Roguelike » dans lequel le joueur devra naviguer un donjon afin d'en récupérer les trésors tout en évitant les monstres qui s'y trouve.

À compléter

Pour compléter ce projet, vous devez créer et soumettre un site Web valide qui comprend le HTML, CSS et Javascript. Le projet sera un petit jeu communément appelé « Dungeon Crawler », dans lequel le joueur contrôlera le héros en tentant d'éviter les monstres et de collecter tous les trésors du donjon. Si le joueur collecte tous les trésors du donjon il gagne et s'il se fait attraper par un monstre, il perd et doit recommencer.

Assurez-vous de bien lire ce document et de respecter les consignes pour avoir tous les points.

Règles générales du projet

- Le projet se fait en équipe de 2 ou seul.
- Votre site Web ne peut pas être basé sur un gabarit existant ou sur une copie d'un autre site Web. Si un tel site Web est soumis, une note de 0 sera attribuée.
- Vous pouvez vous aider de ressources en ligne pour résoudre certains problèmes, mais le plagiat est bien sûr interdit et toute forme de plagiat se verra attribuer la note de 0. Si vous utilisez une petite partie de code d'un autre site, vous devez l'indiquer en commentaire.
- Assurez-vous d'utiliser **les liens relatifs** (`href="/img/image.jpg"`) lors de l'inclusion d'images (``) ou de liens web (`<a>`) plutôt que les liens absolus (`href="C:/Users/X/Projet/img/image.jpg"`). Sinon vos images et liens ne fonctionneront pas lors de l'évaluation.
- Assurez-vous de bien commenter votre code.

Règles spécifiques du projet

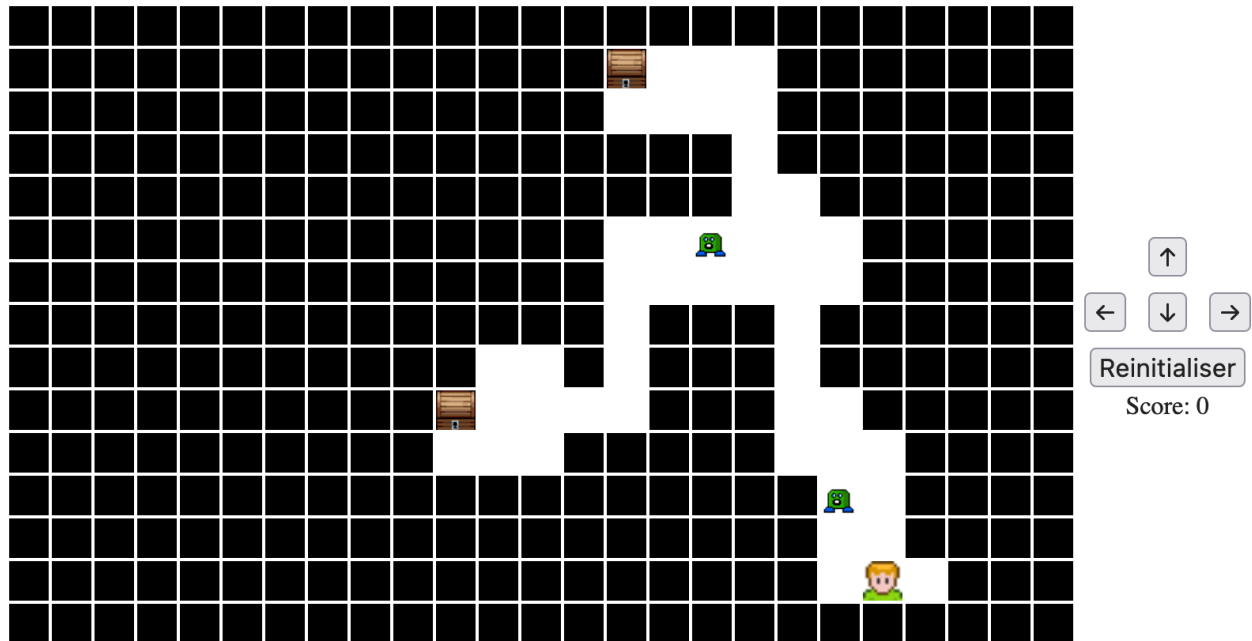


Image donnée à titre indicatif seulement, votre projet peut être différent.

Grille du donjon

La grille du donjon sera représentée par un tableau à 2 dimensions et devra être **minimalement** d'une largeur de 25 cellules et d'une hauteur de 15 cellules.

Chaque cellule pourra comprendre soit :

- Le joueur
- Un monstre
- Un trésor
- Une cellule vide (Le sol)
- Un mur

La disposition de la grille et des objets sur celle-ci sera préconçue par vous, les développeurs, et sera toujours la même.

Joueur

Le personnage commencera à l'endroit que vous aurez choisi dans votre donjon. Chaque tour, le joueur choisira une action (haut, bas, gauche, droite) et le jeu déplacera le joueur dans la direction choisie sur la grille. Bien sûr, le joueur ne peut pas traverser les murs ou bien aller en dehors de votre grille, il faudra donc vous assurez de vérifier ces conditions avant de le déplacer.

Monstres

Les monstres commencent aussi aux endroits que vous aurez prédéterminés sur votre grille (**minimum 2 monstres**). À chaque action du joueur (déplacement), chacun des monstres se déplacera aléatoirement dans une direction : gauche/droite ou haut/bas. Vous pouvez choisir seulement l'une des deux options, donc le monstre se déplace aléatoirement entre gauche/droite ou aléatoirement entre bas/haut.

Si le joueur tombe sur une même cellule qu'un monstre, la partie est terminée : le joueur ne peut plus bouger et il doit recommencer. Vous devez donc avoir un bouton pour permettre au joueur de recommencer à zéro.

Trésors

Les trésors seront placés à des endroits prédéterminés sur votre grille (**minimum 2 trésors**). Si le joueur tombe sur la même cellule qu'un trésor, le score augmentera de 1 point et le trésor disparaît. Si le joueur obtient tous les trésors de la grille, la partie est gagnée et le joueur ne peut plus bouger.

Affichage

Vous pouvez utiliser les images que vous voulez ou même seulement des couleurs pour l'affichage du donjon et de la grille, mais tout doit être clair à l'écran. On doit pouvoir distinguer les monstres, le joueur, les trésors, les cellules vides et les murs. Le score doit être affiché à l'écran.

Résumé

Voici un récapitulatif des contraintes spécifiées dans les paragraphes précédents :

- La grille est représentée par un tableau en 2 dimensions avec une taille minimale de 25 cellules de large et 15 cellules de haut.
- La grille est préconçue et peut avoir les éléments suivants : joueur, trésor, mur, vide (sol), monstre.
- Le joueur doit pouvoir se déplacer dans les 4 directions et il ne peut pas se déplacer sur un mur ou à l'extérieur de votre grille.
- Minimum 2 monstres.
- Les monstres sur votre grille se déplacent automatiquement dans une direction aléatoire à chaque fois que le joueur clique sur un bouton.
- Si le joueur tombe sur une même cellule qu'un monstre, la partie est terminée, il ne peut plus bouger.
- Il doit y avoir un bouton pour permettre au joueur de recommencer la partie.
- Minimum 2 trésors.
- Si le joueur tombe sur la même cellule qu'un trésor, son score est augmenté de 1 et le trésor enlevé.

- Lorsqu'un joueur obtient tous les trésors, il a gagné la partie, ne peut plus bouger et doit recommencer.
- Vous pouvez prendre l'affichage que vous voulez à l'écran, mais l'affichage de tous les éléments (joueur, trésors, monstres, murs, sol, score) doit être clair et on doit pouvoir distinguer chaque élément.
- Assurez-vous que votre code est clair et bien commenté.

Fonctionnalités supplémentaires

Les fonctionnalités décrites ci-dessus sont les fonctionnalités qui sont requises pour obtenir tous les points. Cependant, si vous en êtes capables, je vous encourage fortement à vous lancer un défi et à implémenter des fonctionnalités supplémentaires! J'attribuerai des points bonis jusqu'à un **maximum** de 20% pour ces fonctionnalités en fonction de leur complexité.

Voici quelques exemples en ordre de la difficulté que je pense chaque fonctionnalité est à implémenter :

- Ajouter d'autres types de trésors qui ajoute plus de points que seulement 1. (5%)
- Ajouter un écran de fin de partie qui dira au joueur son score et lui permettra de recommencer la partie. (5%)
- Ajouter une barre de vie pour le joueur et permettre au personnage d'être attaqué par un monstre plusieurs fois avant de perdre la partie. (5%)
- Ajouter d'autres types d'ennemis qui font autre chose que de terminer la partie, par exemple qui empêcherait le joueur de se déplacer pour un tour. (10%)
- Ajouter des pièges qui s'active/se désactive à chaque tour. (10%)
- Permettre au joueur d'attaquer (se débarrasser des monstres) avec un autre bouton lorsqu'il est à côté d'un monstre. (10%)
- Implémenter une génération procédurale du donjon. C'est-à-dire que tous les murs, le sol, l'emplacement des trésors, joueur, monstres, etc. est aléatoire à chaque partie. (20% et mon respect 😊)
- Autre (À voir avec le professeur)

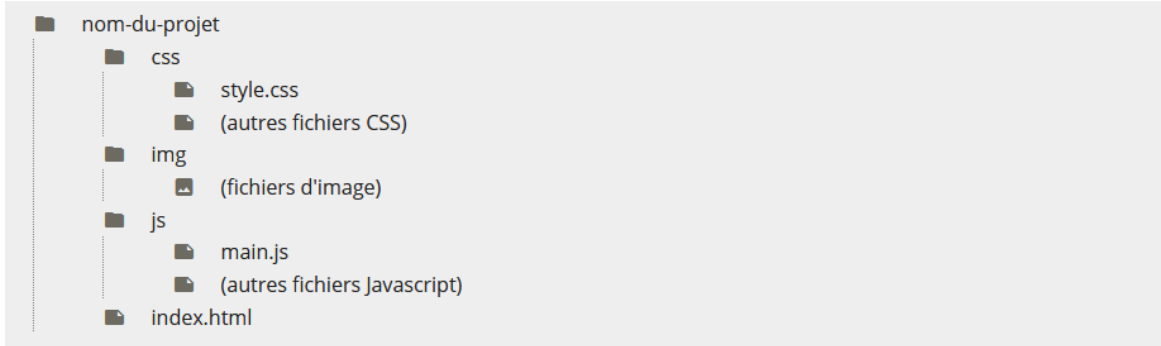
Si vous ajoutez des fonctionnalités supplémentaires, assurez-vous de garder le jeu équilibré pour qu'il ne soit pas trop facile ou impossible à terminer.

Soumission

La date de soumission du laboratoire est indiquée sur eCité. Vous devez soumettre votre projet sur Github tel que vu en classe en respectant la structure ci-bas. Vous n'avez qu'à inscrire le lien vers votre site dans le pigeonier sur eCité.

Après la date de soumission, le dernier commit de votre site Web fait sur Github sera considéré comme votre soumission. Aucun retard ne sera accepté.

Assurez-vous de mettre votre repository publique et de publier votre site web! Si vous ne vous rappelez plus comment faire, une vidéo démonstrative est disponible sur eCité.



Liens et outils

Si vous avez besoin d'images ou d'icônes, vous pouvez utiliser celles se trouvant aux liens ci-dessous. Assurez-vous de toujours utiliser des images dont vous avez les droits lorsque vous produisez un site Web. Assurez-vous de lire les licences d'utilisation.

- [Unsplash](#)
- [Pexels](#)
- [Pixabay](#)
- [Iconmonstr](#)
- [OpenGameArt](#)

Grille d'évaluation

Description		Poids
HTML		/10
	Code HTML respecte les consignes vues en classe et ne contient pas d'erreurs (Validation W3C)	/10
CSS		/10
	Code CSS respecte les consignes vues en classe et ne contient pas d'erreurs (Validation W3C)	/10
Javascript		/75
	Fonctionnement du jeu et respect de toutes les contraintes	/40
	Code Javascript respecte les bonnes pratiques vues en classe, ne contient pas d'erreurs et bien commenté	/10
	Fonctions Javascript bien définies	/10
	Utilisation de tableau(x) et/ou objets pour le stockage d'informations de façon appropriée	/5
	Manipulation du DOM faite de façon appropriée	/10
Général		/10
	Présentation/Affichage clair et propre	/10
Total		/105 (35%)
Bonus	Fonctionnalités supplémentaires (max 20%)	/21

Conseils

Lorsque l'on commence un projet de cette taille, cela peut être intimidant et l'on peut ne pas savoir par où commencer. Voici quelques conseils pour vous aider :

- Ne commencez pas par l'interface graphique. Dans ce genre de programme, il est plus facile de s'occuper des données en mémoire et de leur modification. Par exemple, commencez par le stockage de la grille et des différents éléments sur celle-ci, puis modifiez l'interface graphique uniquement lorsque les opérations en mémoire fonctionnent bien.
- Le donjon est une grille à 2 dimensions. Il doit donc être un tableau à 2 dimensions en mémoire. Ce tableau contiendra des valeurs indiquant ce qu'il y a pour chaque tuile du donjon, soit une tuile vide (sol), un mur, un monstre, un trésor ou le joueur. Ces valeurs peuvent être des entiers (0, 1, 2, 3, ...), des caractères ou même du texte. À vous de décider comment vous voulez le programmer.
- Séparez certains concepts dans différents fichiers. Il est plus facile de comprendre le code et de bien s'organiser si vous séparer les différents éléments de votre jeu dans différents fichiers Javascript. Dans notre cas, on pourrait probablement séparer la grille du donjon, le joueur et la logique principale du jeu dans leur propre fichier. N'hésitez pas à vous

créer plus de fichiers si c'est nécessaire et n'oubliez pas que chaque fichier est exécuté dans l'ordre que vous l'avez mis dans votre html.

- Commencez par faire une interface simple. L'affichage dans le navigateur peut simplement être une table avec différentes couleurs pour identifier le joueur, les monstres, etc. Rien ne vous empêche par la suite d'améliorer votre interface.