Tower of Power

# Mise en situation

La compagnie CVMco contacte votre équipe afin de vous offrir un contrat de réalisation pour le site web d’un jeu sur navigateur (browser-based game). Vous apprenez que le contrat avait préalablement été accordé à une autre équipe, mais, insatisfaite de la progression du projet, CVMco a racheté le contrat, conservant le code initial. Ce code est à votre disposition. La seule exigence de CVMco est que les pages du site doivent conserver une apparence commune entre elles. Cependant, vous n’avez pas à conserver l’apparence héritée du code initial, en fait, vous devez la modifier.

# Contraintes de jeu

Le jeu que vous allez réaliser doit posséder les caractéristiques suivantes :

* Le joueur doit pouvoir sélectionner un protagoniste d’une liste offrant au moins 4 choix différents.
* Jeu de type platformer, genre Castlevania, disposé en tour immense comportant au moins 3 niveaux aux caractéristiques visuelles distinctes.
* La fin de chaque niveau de la tour est guettée par un « big boss » qui doit être vaincu.
  + L’objectif du jeu est de vaincre le « big big boss » du dernier étage.
* L’organisation spatiale doit être disposée en tuiles carrées de 32X32 pixels.
* Chaque niveau doit avoir une largeur d’au moins 200 tuiles avant de monter au niveau suivant.
* À chaque niveau, il y a au moins 15 ennemis à vaincre et 3 objets à collectionner pour aider le joueur.
  + Les objets à collectionner doivent être pertinents :
    - Arme pour vaincre le « big boss »
    - Clé pour ouvrir une porte vers un endroit caché ou un autre niveau
    - Chandelle ou lampe de poche pour voir dans le noir
    - Etc.
* Vous devez gérer la physique (gravité, collisions, vent, etc).
* Il doit y avoir 2 save-points par niveau : un au début, et un au milieu.
  + Le progrès ET les objets collectionnés doivent être emmagasinés dans la base de données et rechargés lorsque le joueur reprend la partie au dernier save-point.
* De la musique et des effets sonores sont obligatoires.

Hormis ces contraintes, votre équipe a la liberté de choisir le ou la protagoniste comme bon vous semble. La musique et le visuel relèvent également de votre choix. Vous avez aussi la liberté de développer l’histoire et le déroulement du jeu selon vos goûts (tant qu’ils soient de bon goût).

# Contraintes pour le site web

L’organisation du site web doit respecter les exigences énumérées ci-dessous. On doit pouvoir naviguer à n’importe quelle de ces pages à tout moment. Voici les pages qui DOIVENT exister :

* Accueil avec login/signup
* À propos
* Lobby du jeu
  + C’est d’ici qu’on démarre une partie. On peut :
    - Sélectionner une nouvelle partie ou une partie existante
      * Si c’est une nouvelle partie, choisir l’avatar voulu.
      * Un utilisateur ne peut avoir qu’une seule partie de sauvegardée à un moment donné.
        + On doit afficher des statistiques de progression.
* Palmarès pour tous les joueurs
* Commentaires
  + Il doit y avoir un bouton pour ajouter des nouveaux commentaires.
    - Tous les commentaires doivent exister dans la BD.
  + Pour chaque commentaire, on affiche :
    - Nom d’utilisateur
    - Date du commentaire
    - Bouton pour éditer
    - Bouton pour retirer

# Contraintes de procédure

Toujours dans l’intérêt de protéger leur investissement, CVMco exige que, dans la mesure du possible, vous vous conformiez à la méthode [Scrum](https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement)), tout en respectant la [Manifeste Agile](https://fr.wikipedia.org/wiki/Manifeste_agile). Votre équipe entière doit donc se rencontrer au moins une fois par semaine lors des heures de cours afin d’inspecter le code ensemble et de veiller au progrès du projet. Nous référerons à ces rencontres à l’aide du terme *scrum* (*mêlée* si vous préférez). De plus, à chaque 3 ou 4 semaines, un représentant de la compagnie (le professeur) vous rencontrera pour évaluer votre progrès. Ces périodes de 3 à 4 semaines sont nommées des *sprints*. Le représentant examinera votre code et exigera les documents suivants :

* Sprint 0 (4 semaines 🡪 25 septembre)
  + CVMco souhaite voir cette documentation AVANT que vous écriviez du code, question de corriger le tir au besoin en amont plutôt qu’en aval.
    - Proposition (500 mots)
      * Après avoir digérer le mandat (le présent document), vous devez rédiger une proposition de tâche à CVMco.
    - Diagramme de cas d’usage
      * Utilisez un logiciel de rendu visuel pour créer le diagramme de cas d’usage (use case) en UML.
    - Diagramme de classe
      * Utilisez un logiciel de rendu visuel pour créer le diagramme de classe en UML.
    - Modèle relationnel
      * Utilisez un logiciel de rendu visuel pour élaborer le modèle relationnel de votre base de données.
    - Maquettes
      * Utilisez un logiciel de rendu visuel pour faire des ébauches graphiques du site web et du jeu.
    - Planification et suivi
      * Le document Excel à remplir et à remettre à chaque sprint vous est fourni par CVMco.
* Sprint 1 (3 semaines 🡪 23 octobre)
  + Rencontre routinière
    - Code
    - Document de suivi à remplir
      * Carnet de sprint
* Sprint 2 (3 semaines 🡪 13 novembre)
  + Rencontre routinière
    - Code
    - Document de suivi à remplir
      * Carnet de sprint
* Sprint 3 (3 semaines 🡪 4 décembre)
  + Rencontre finale
    - Code
    - Document de suivi à remplir
      * Carnet de sprint

# Contraintes techniques

* Côté serveur
  + Python, Flask et sqlite (BD)
* Côté client
  + HTML, CSS et Javascript
    - Pas de librairies externes pour le javascript
* Code et documentation
  + Toute votre documentation et votre code devra être partagé aux acteurs principaux du projet (membres de l’équipe et le professeur) à l’aide de [GitHub](https://github.com/login).
    - Le nom du dépôt vous sera fourni en classe.
    - Le dépôt GitHub doit absolument respecter l’arborescence suivante :
      * racine du dépôt
        + .git
        + .gitignore
        + readme.md
        + documentation

Les documents du sprint 0 doivent être ici, ainsi que le document de planification et de suivi.

* + - * + code

L’arborescence entière de votre site doit être ici.

# Suggestions

* Prenez soin de bien partager le travail entre les membres de l’équipes.
* Ne commettez pas l’erreur de créer tout le *back-end* (serveur + BD) avant de commencer à développer le *front-end*.
  + Bien que théoriquement *impossible* selon certains, le développement simultané des côtés client et serveur informera plusieurs aspects du développement du côté serveur.
* Lors du développement du jeu, ne mettez pas trop de temps au début sur l’aspect graphique final plutôt que sur la fonctionnalité du jeu.