

Projet : un générateur de cartes (pour jeu de rôles ou de plateau)

UCBL - Département Informatique de Lyon 1 – BDW1 - automne 2020

- Ce projet est évalué par deux notes : soutenance et réalisation.
- Cette UE est avant tout une introduction aux **bases de données**. La non-utilisation d'une BD relationnelle dans le projet peut se voir attribuer la note 0.
- Le rendu du projet et la soutenance ont lieu lors de la dernière séance de TP. Tout rendu ne respectant pas les consignes peut se voir attribuer la note 0.
- Vu la quantité de travail demandé, il est fortement recommandé de réaliser le projet en binôme (pas de trinôme). Les étudiant.e.s travaillant seul.e sur le projet seront notés comme ceux travaillant en binôme. Un binôme signifie deux étudiant.e.s **du même groupe de TD**.
- Attention à la gestion de votre temps : concentrez-vous sur l'implémentation des fonctionnalités demandées, et pas plus. Cela ne sert à rien de gagner un point bonus pour quelques améliorations mineures mais de perdre plusieurs points pour des fonctionnalités non développées.

Ce jeu de données porte sur la génération de cartes pour des jeux. Quelques exemples de générateurs de carte : [fantasy-map-generator](#), [donjon](#), [gozzys](#), [Dave's mapper](#), [dungeon painter](#) ainsi que des listes plus complètes sur [d&d compendium](#) et [wiki roll20](#).

1 Des spécifications au script SQL

Les [jeux de plateau](#) et [jeux de rôles](#) peuvent nécessiter des cartes (e.g., un champ de bataille, un château en ruines, des planètes à conquérir). Ici, on se place plutôt dans un cadre *fantasy*, et les cartes représentent par exemple des donjons, une cité médiévale ou une forêt enchantée. Une carte possède un nom, une description et une date de création. Elle est générée à partir de plusieurs paramètres (e.g., pourcentage de forêt ou de culs-de-sac, quantité de pièges ou de passage secrets). Chaque paramètre possède un nom et une valeur. La carte peut être générée à nouveau en changeant la valeur de ses paramètres, et on conserve un historique des valeurs précédentes. On stocke le nom, prénom et date d'inscription de contributrices, qui peuvent soit créer une carte ou en modifier une existante. Une seule personne crée une carte, mais il peut y avoir de multiples contributions sur une carte. On conserve d'ailleurs la date et le type (e.g., ajout d'un piège, suppression d'un monstre) de chaque contribution. Une carte se découpe en zones, au minimum une (e.g., une salle, un couloir, une clairière), qui se caractérisent par une description, des dimensions (longueur et largeur, souvent un nombre de cases) et un environnement. Une zone étant très spécifique, elle n'appartient qu'à une seule carte. Une zone est reliée à d'autres, au maximum une pour chacun des quatre points cardinaux : on stocke donc la direction de la salle voisine. Mais on peut également trouver des passages secrets, qui permettent de relier deux zones non adjacentes (sans considérer la direction) par un élément du mobilier. Il n'y a jamais plus de deux passages secrets par zone. Un passage secret possède une (unique) difficulté de détection (e.g., *très facile*, *difficile* voire *impossible*). Dans une zone, on trouve un certain nombre d'éléments fixes, à une position donnée dans la zone, et qui possèdent un nom et un chemin vers une image les représentant. Une première catégorie d'éléments est le mobilier (au sens large, e.g., des tables, des tonneaux mais aussi des rochers ou des buissons en extérieur), qui peut être déplaçable ou pas et qui occupe un certain nombre de cases. Les équipements ont simplement une valeur monétaire (en pièces d'or) et n'occupent qu'une seule case. La troisième catégorie, appelée piège, inclut une catégorie, une zone d'effet, et trois difficultés (de détection, d'esquive et de désamorçage). La zone d'effet correspond au nombre de cases occupées par le piège. Une zone peut aussi être source de rencontres, soit avec des personnages non joueurs (PNJ) et des créatures. Chaque être rencontré est doté d'une quantité de pièces

d'or, de points d'attaque et de points de vie. Pour simplifier, tous les êtres n'occupent qu'une seule case, même le terrifiant Jabberwocky. Le rôle des **PNJ** est de guider les joueuses, de leur donner des quêtes ou des indices, ou simplement pour étoffer l'histoire de menus détails. On leur attribue un métier, un trait de caractère et une phrase type de dialogue. Au contraire, les créatures, caractérisées par un climat et un niveau de difficulté, sont les ennemis des joueuses. Une créature peut posséder un environnement principal et des environnements secondaires où on le rencontre (e.g., aquatique, jungle, montagne). Cela peut permettre de connaître les points forts ou faibles d'un ennemi. En plus du nom, une description accompagne généralement chaque environnement. Pour gagner une carte, il existe deux types d'objectifs : atteindre une zone ou trouver un équipement. Une carte ne possède qu'un seul objectif.

Dans un premier temps, nous allons concevoir la base de données à partir des spécifications. Votre modélisation doit respecter au mieux ces spécifications, mais votre site web n'utilisera pas tous les concepts décrits.

1. Produisez un diagramme entité / association pour ces spécifications (par exemple avec [AnalyseSI](#), [MoCoDo](#), [Looping](#) ou [JMerise](#)).
2. Produisez le schéma relationnel dérivé de votre diagramme E/A. Si vous générez ce schéma avec un outil de modélisation, il est recommandé de le vérifier et éventuellement de le corriger / compléter.
3. Produisez le script SQL permettant la création de la base de données. Si vous générez ce script avec un outil de modélisation, il sera nécessaire de le vérifier et de le corriger / compléter (ces modifications devraient être stockées dans un autre fichier que celui généré par l'outil de modélisation).

2 Design du site et pages statiques

Vous êtes libres d'organiser votre site comme bon vous semble. Mais chacune de vos pages doit avoir les cinq zones suivantes :

- Un entête (**<header>**), avec un logo (au choix, mais cliquable pour revenir à l'accueil) et un nom de site ;
- Un menu (**<nav>**), dont les libellés seront explicites ;
- Des statistiques sur la base de données : nombre d'instances pour les tables importantes, nombre de cartes générées par contributrice, top-3 des pièges les plus utilisés, etc. ;
- Le contenu de la page, qui correspond aux fonctionnalités développées dans les sections suivantes. Prévoyez également de créer une page d'accueil qui décrit les objectifs de votre site ;
- Un pied de page (**<footer>**), avec un lien vers le site de l'UCBL, un autre vers la page du BDW1, l'année courante et des remerciements (e.g., site [OpenClipart](#) pour les images).

Il ne faut **pas implémenter un système d'authentification**¹ ! La mise en page et mise en forme se feront évidemment avec des styles CSS. L'esthétique est prise en compte lors de la notation, aussi soignez votre site. Le projet sera évalué avec le navigateur **Mozilla Firefox**, donc vérifiez le rendu de votre site avec ce navigateur !

3 Fonctionnalité 1 : intégrer et afficher les données

Cette fonctionnalité consiste à migrer les données existantes fournies (créatures, pièges et mobilier²) dans votre base de données, puis les afficher sur une page. L'utilisation de ces données est **obligatoire**. Ces données se trouvent dans la base de données partagée **dataset**, qui contient une unique table **DonneesFournies**. Elle n'est malheureusement pas bien modélisée, et il est donc nécessaire de stocker son contenu dans votre base. C'est un processus fréquent appelé *intégration de données*.

Vous allez donc écrire un script PHP qui interroge la base **dataset** pour extraire les informations pertinentes, les transformer, puis les insérer dans votre base. En théorie, ce script ne s'exécute qu'une seule fois. Mais en pratique, vous allez le tester à plusieurs reprises. N'oubliez pas de vider votre base de données avant de le ré-exécuter !

¹Un système d'authentification se base sur des protocoles standardisés et sécurisés comme SSL, Kerberos ou CAS.

²Les données proviennent de différentes sources, notamment [pathfinder-fr](#) et [naheulbeuk](#).

On peut décomposer l'étape d'intégration en trois étapes :

- Réfléchir aux correspondances entre les données existantes et votre schéma (e.g., à quel attribut de votre schéma correspond tel attribut fourni). Vérifier si les types de données sont cohérents entre attributs correspondants, et quelles sont les transformations nécessaires au niveau des valeurs. N'oubliez pas que le jeu de données fourni n'est pas bien modélisé ;
- Écrire une requête **select** pour récupérer les informations pertinentes. Pour requêter, il faut indiquer le nom de la BD avant la table : **SELECT * FROM dataset.DonneesFournies** ;
- Pour chaque tuple résultat, appliquer éventuellement des transformations sur les valeurs (soit directement en SQL, soit en PHP), puis écrivez une requête **insert** pour peupler vos tables. Des fonctions PHP comme [explode](#) et [parse_str](#) peuvent être utiles. N'oubliez pas de peupler les tables intermédiaires si besoin !

Avant la soutenance, il est possible que l'on supprime la base de données **dataset**. Si votre intégration est réussie, vous disposez normalement de *887* créatures, *11* environnements, *193* pièges et *27* éléments de mobilier.

Une fois que les données fournies sont dans vos tables, écrivez une page PHP pour les afficher. Un formulaire permet de choisir le type de données à afficher (créatures, piège ou mobilier) et sa validation entraîne l'affichage des données du type choisi dans un tableau (avec autant de colonnes que d'attributs disponibles).

Il est fortement recommandé d'utiliser PHPMYAdmin pour insérer des instances fictives dans les autres tables (e.g., des PNJ, des équipements, davantage de mobilier, quelques contributrices) en prévision des tests fonctionnels et de la soutenance.

4 Fonctionnalité 2 : générer une zone

Pour cette deuxième fonctionnalité, nous allons créer une page pour générer aléatoirement une zone. Celle-ci inclut un formulaire qui configure les paramètres de la zone à produire, et cette dernière sera affichée sous le formulaire et stockée en base.

Ci-dessous la liste des paramètres :

- Saisie d'une description et des dimensions de la zone, et choix d'un environnement (parmi ceux disponibles) ;
- Nombres minimal et maximal d'équipements, de pièges, de mobilier, de créatures et de PNJ. Il faut vérifier que le total n'excède pas le nombre de cases disponibles (en fonction des dimensions de la zone), et ce n'est pas possible de superposer des éléments ;

Pensez à proposer des valeurs par défaut pour chaque paramètre, afin de ne pas avoir à remplir tous les champs du formulaire lors des tests. Votre script doit ensuite créer la zone selon les valeurs de ces paramètres, en sélectionnant un nombre aléatoire d'instances dans les intervalles spécifiés.

Au niveau de l'affichage, la zone sera représentée par un tableau (nombre de cases correspondant aux dimensions de la zone). Les autres objets (éléments fixes ou rencontres) seront placés dans les cases du tableau (par exemple sous forme d'une lettre ou d'une image). Il faut prévoir d'afficher les informations de ces objets (e.g., avec une légende à côté, ou dans une infobulle). Vous êtes évidemment libres d'ajouter d'autres paramètres ou d'améliorer l'affichage (e.g., en utilisant de vraies tuiles et des images pour les objets).

Lors de la soutenance, il est fort probable que les enseignant.e.s testent votre application en saisissant des valeurs absurdes susceptibles de déclencher des erreurs dans votre application. Alors essayez de penser au pire !

5 Fonctionnalité 3 : générer une carte

Cette fonctionnalité nécessite de réutiliser partiellement la précédente (génération d'une zone) afin de générer une carte complète. Comme précédemment, la page comportera un formulaire pour renseigner les informations sur la carte ainsi que ses paramètres.

Ci-dessous la liste des paramètres :

- Saisie d'un nom, d'une description, et d'une créatrice (parmi les contributrices déjà stockées).
- Nombres minimal et maximal de zones, et dimensions minimale et maximale des zones. Chaque zone aura une description générée (e.g., zone 1, zone 2) et des dimensions choisies aléatoirement dans l'intervalle spécifié ;
- Choix de plusieurs environnements pour la carte. Une zone aura un environnement choisi aléatoirement parmi ceux de la carte ;
- Nombres minimal et maximal d'équipements, de pièges, de mobilier, de créatures et de PNJ (chaque zone aura une valeur choisie aléatoirement dans ces intervalles, tout en vérifiant que le nombre de cases occupées soit bien inférieur aux dimensions de la zone) ;
- Choix d'un type d'objectif (équipement ou zone) ;
- Nombre moyen de zones reliées à une zone donnée, ainsi que le nombre de passages secrets dans la carte.

Quand une carte est générée, on la stocke (avec ses paramètres initiaux) et on l'affiche sous le formulaire. Il est alors possible de modifier les paramètres afin de produire une nouvelle version de la carte : les zones précédemment générées sont alors supprimées de la base, mais on conserve l'historique des précédentes configurations.

Certaines informations sont générées à la fin. Par exemple, pour relier les zones entre elles, il faut évidemment avoir déjà généré chaque zone. De même pour renseigner l'équipement ou la zone qui sert d'objectif à la carte.

Si vous avez le temps, vous pouvez adapter l'algorithme de génération de carte (e.g., afin de pouvoir conserver certaines zones quand on génère une nouvelle version).

6 Fonctionnalité 4 : au choix

Vous êtes libres de choisir la dernière fonctionnalité que vous développerez. Celle-ci doit être conséquente (i.e., pas un simple affichage de données - vérifiez auprès de vos enseignants) et doit être présentée clairement sur vos transparents. Vous pouvez adapter ou compléter les règles du jeu (ex, pour gérer les pièges ou les combats). Quelques exemples de fonctionnalités :

- Production de jolies cartes, dont les zones et leurs éléments s'agencent bien, avec des graphismes variés, éventuellement une musique d'ambiance, etc. Nombreuses ressources sur [OpenGameArt](#), [naheulbeuk](#) (notamment le pack de dalles carrées), [les tuiles de Hero Quest](#), [table top audio](#), etc. ;
- Création de personnages (pour parcourir la carte générée) avec choix d'un portrait et/ou icône, génération de statistiques, personnalisation de la description et autres détails ;
- Exploration automatique de la carte avec un personnage généré aléatoirement (i.e., c'est le programme qui déplace et agit pour le personnage). Ce dernier réalise plusieurs actions comme le déplacement, la fouille (de passages secrets, d'équipements ou de rencontres), le combat. Bien penser à détecter une situation de blocage ou de victoire, et réfléchir à la façon avec laquelle l'utilisateur suit la progression du personnage à l'écran ;
- Exploration interactive de la carte avec un personnage généré aléatoirement. Cette fois, le ou la joueuse décide des actions que son personnage réalise. Là aussi, bien réfléchir à la manière de faire avant de se lancer (une interaction clavier/souris nécessite de recharger la page - à moins d'utiliser du JavaScript).

7 Préparation des livrables

Trois livrables sont à rendre **le jour de soutenance de votre projet, avant minuit, sur Tomuss** :

- Une archive contenant votre site web, en zip ou rar (colonne *archive_projet*)
- L'URL de votre site, sur le serveur `bdw1.univ-lyon1.fr` (colonne *url_projet*)
- Les transparents de soutenance en pdf (colonne *transparents_projet*)

7.1 Archive avec le site web

L'archive est au format **zip** ou **rar**, et respecte les contraintes suivantes :

- Une **seule archive par binôme**
- L'archive contient au minimum **4 fichiers** :
 - le répertoire contenant les pages de votre site (code commenté et indenté). La page *index* contiendra vos noms et numéros étudiant en commentaire ;
 - un fichier **.txt** ou **.sql** avec le script SQL **"exécutable"** de création de votre base de données ;
 - deux images (ou PDF), l'une de votre diagramme E/A et l'autre de votre schéma Relationnel.

7.2 URL du site web

Après le rendu du projet, nous testerons votre application en utilisant l'URL saisie dans Tomuss :

- Elle doit être complète (login et répertoire(s), voire fichier index précisé si nécessaire) et elle doit pointer vers la page d'accueil de votre site (e.g., http://bdw1.univ-lyon1.fr/p1234567/mon_site/). Merci de la vérifier une fois qu'elle est saisie dans Tomuss !
- Votre site doit déjà avoir des instances stockées en base **en nombre suffisant** pour simplifier les tests.

7.3 Soutenance

La soutenance dure 20 minutes par binôme et se décompose en deux parties : une présentation avec démo et avec transparents (8 minutes) et une séance de questions (12 minutes). Les conditions suivantes s'appliquent :

- Pour la partie présentation, respectez le temps ! Vous serez interrompus au bout des 8 minutes, et donc pénalisés.
- Les deux membres du binôme doivent parler lors de la présentation. Entraînez-vous.
- Vous présentez soit sur votre machine, soit sur une machine de la fac. Dans tous les cas, vérifiez avant que la machine permette de lire vos transparents, de faire la démo, etc. Prévoyez d'arriver quelques minutes en avance devant la salle, et quand on vous fait entrer, préparez-vous pour la présentation et les questions (i.e., lancement du site dans le navigateur, ouverture des transparents, ouverture des fichiers source et des diagrammes).
- Vous devez répondre à quatre questions minimum pendant les 12 minutes. Donc donnez des **réponses claires et concises** (pas de baratin). Si vous ne savez pas répondre à une question, dites-le pour ne pas perdre de temps.
- Si vous arrivez en retard à votre soutenance, vous passez en fin de séance... ou pas du tout selon les disponibilités de votre jury.

Vous pouvez réaliser vos transparents avec n'importe quel logiciel (e.g., Libre Office, Power Point, Latex Beamer). Vous devrez par contre **déposer un PDF de vos transparents** sur Tomuss. Les conditions suivantes s'appliquent pour les transparents :

- Les transparents seront soignés et numérotés. Évitez les longues phrases !
- Vos 5 transparents doivent respecter le plan suivant :
 - Transparent 1 : vos noms, prénoms, numéros étudiant, et si vous participez au concours, le nom et/ou le logo de votre site ;
 - Transparent 2 : réalisations (fonctionnalités implémentées, dont description de la fonctionnalité 4) ;
 - Transparent 3 : difficultés rencontrées au niveau technique, organisationnel, etc. (évitez l'éternel "manque de temps") ;
 - Transparent 4 : diagramme E/A (prévoir un agrandissement si c'est illisible) ;
 - Transparent 5 : une capture d'un écran de votre site, qui permet d'annoncer la démo.

- La démo de votre site doit être préparée (déroulement d'un scénario). Vous avez 8 minutes pour les transparents et la démo, donc focalisez-vous sur les points importants.

Concours (optionnel) : Pour vous détendre (en cas de blocage sur un bug, d'engueulade avec votre binôme, etc.), deux concours sont organisés, celui du **meilleur nom de site** et celui du **plus beau logo**. Les deux binômes gagnants remporteront un paquet de chamallows. Et pour ce sujet, un troisième concours sur **la plus jolie carte (générée aléatoirement !)**, à envoyer par mail au responsable de l'UE. Soyez créatifs/ves !

