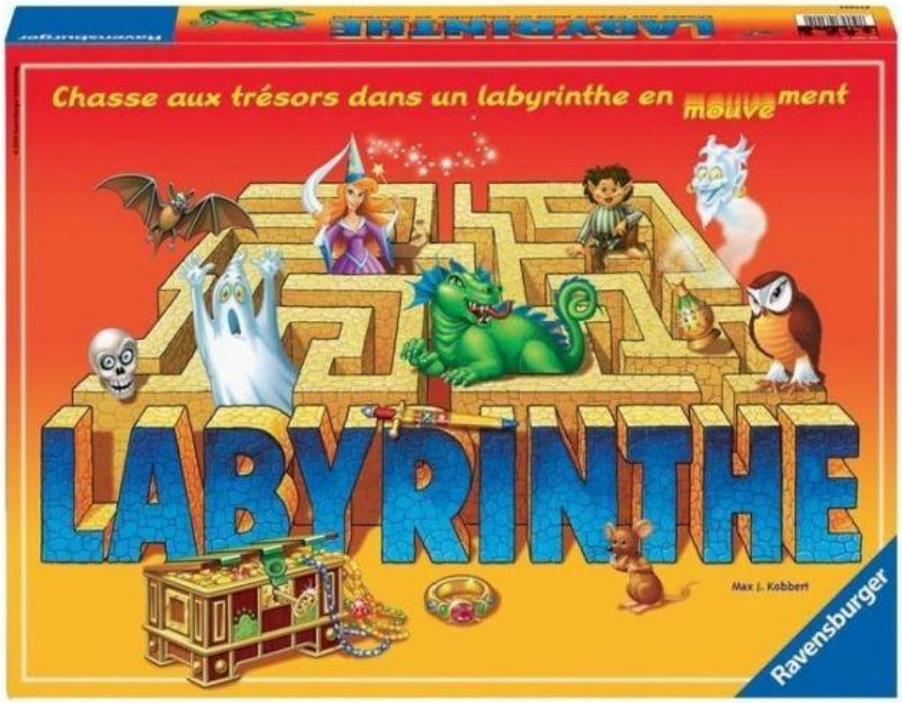
Sujet du projet Informatique semestre 1 - 2017-2018



Le sujet

Contexte : Nous sommes une société dont le but est de rendre « on-line » des jeux de plateaux existants.Nous voulons mettre en place le jeu de plateau "Labyrinthe" en version « on-line » multi-joueurs.

Nous voulons également deux applications finales :

- Une application Web en ligne réalisée en Struts 2
- Une application client lourd en JavaFx

Attention, des joueurs utilisant une application pourront très bien rencontrés des joueurs utilisant l'autre application. Une SFD est donnée à la suite de la présentation du jeu.

A propos du jeu

Le **but du jeu** est d'être le premier à récupérer tous les trésors initialement attribués et à être retourné à sa case de départ.

Contenu du jeu :

Un plateau contenant 16 cases inamovibles (voir photo ci dessous):

- 4 cases couleurs (les cases de départ et d'arrivée pour chacun des joueurs)
- 12 cases trésors citées en partant du coin supérieur gauche au coin inférieur droit de la gauche vers la droite
    - le livre
    - le sac d'or
    - la carte au trésor
    - la couronne
    - le jeu de clés
    - les ossements
    - la bague
    - le coffre au trésor
    - la pierre précieuse (émeraude)
    - l'épée
    - le chandelier
    - le heaume

Toutes les autres cases sont amovibles

- 12 lignes droites dépourvues de trésors
- 10 virages dépourvus de trésors
- 6 virages contenant les trésors suivants : salamandre, chouette, rat, "papillon", scarabée, araignée
- 6 T contenant les trésors suivants : personnage bizarre (troll), chauve-souris, dragon, fantôme rieur, fée, fantôme avec son boulet

Au final, nous avons donc 50 cases : 49 constituant un labyrinthe de 7x7 et une carte amovible à l'extérieur du labyrinthe.

Il y a également 24 cartes trésors répertoriant tous les trésors mentionnés précédemment (cases amovibles ou non).

Préparation du jeu

Les cases amovibles du labyrinthe sont réparties aléatoirement sur les emplacements possibles du plateau. Il reste alors une case amovible non plaçable.

Les joueurs ont leur pion de couleur sur leur case de départ respective. Les 24 cartes "trésors" sont distribuées face cachée entre les joueurs.

Déroulement du jeu

Le déroulement du jeu est précisé dans les règles fournies dans la section suivante.

Documents complémentaires

Vous trouverez ci-dessous quelques informations complémentaires pour la conception du jeu

- Les règles du jeu contenant une description précise du déroulement du jeu : règle

Planning des rendu

24 novembre 2017 à 20h : **LIVRAISON DU MODELE DE L'APPLICATION**

22 décembre 2017 à 20h : **LIVRAISON DE L'APPLICATION STRUTS 2**

17 janvier 2018 à 20h : **LIVRAISON DE L'APPLICATION JAVA FX**

18 et 19 janvier 2018 : SOUTENANCES

Spécification fonctionnelle détaillée

Au delà du fait que les règles du jeu doivent être évidemment respectées, nous vous demandons les fonctionnalités suivantes :

- Système d'authentification - Le joueur s'inscrit et choisit un login/mot de passe. Si ce login est disponible, alors il parvient à se connecter. Dans le cas contraire, on lui demande de saisir un autre login.
- Un joueur connecté peut quitter la plate-forme et se déconnecter. Un joueur connecté inactif sera déconnecté automatiquement au bout de deux minutes.
- Une fois le joueur inscrit puis authentifié, il arrivera sur une fenêtre lui affichant la liste des joueurs et des parties en cours.
- Un joueur peut créer une partie publique ou une partie privée en spécifiant le nombre de joueurs attendus (de 2 à 4 le créateur inclus).
- Dans une partie publique, tout joueur connecté (qui n'est pas déjà dans une partie) peut rejoindre cette partie.
- Tout joueur peut rejoindre une partie publique en cours en tant qu'observateur uniquement. Il voit alors le plateau etvoit l'état de la partie (les trésors découverts par chaque joueurs).
- Dans une partie privée, le joueur qui a créé la partie invite un autre joueur à y participer. Il peut lancer plusieurs invitations. Une fois le nombre de joueurs espérés atteint (nombre d'invitations acceptées + le créateur = nombre de joueurs attendus), la partie est lancée.
- Un joueur ne peut pas être inactif plus de 15 secondes lors de son tour de jeu. Si le délai est dépassé sans aucune activité avec le jeu, il passera son tour et ne se déplacera pas.

Contraintes techniques

- Une partie devra être possible entre des joueurs utilisant les deux plates-formes. Vous pourrez mettre en place un RMI serveur hébergé sur le serveur Tomcat pour que l'application JavaFX se connecte au jeu et utilise ainsi la même instance du modèle. Un exemple vous est fourni à la fin de ce sujet.
- On ne demande pas d'utiliser une base de données. Les logins utilisés à un instant t sont uniques et sont perdus lors du redémarrage du serveur.
- Votre projet devra être organisé d'un projet principal (parent) et de trois modules (modele, appliStruts et appliJFX). Une organisation similaire est retrouvée dans l'exemple fourni dans la dernière section.
- Les 3 projets aboutissant à des livraisons doivent être des projets Maven. Des détails sur la méthodologie de livraison vous seront donnés plus tard.
- Votre projet parent devra être identifiable par maven de la façon suivante : fr.miage.orleans.projet:labyrintheGroupe? où ? est la lettre de votre groupe (groupid:artifactId)
- Votre modèle devra être identifiable par maven avec l'artefact id suivant : modele
- Votre application Struts devra être identifiable par maven avec l'artefact id suivant : appliStruts
- Votre application JFX devra être identifiable par maven avec l'artefact id suivant : appliJFX

Matériel à votre disposition :

- un repo Git
- les analyses Sonar
- Exemple permettant de lancer un serveur RMI sur un serveur Tomcat : here!
  Ce sont trois projets différents. Cependant les dépendances sont définies dans le pom.xml de chaque projet. Ainsi, il vous faut faire un "install" du modèle (PopulationRMI) pour que les autres puissent le récupérer. Une fois fait, vous pouvez mettre à jour les deux autres projets en cliquant "droit" sur le projet puis en sélectionnant Maven > Reimport. Normalement chaque projet devrait récupérer le modèle que vous avez installé. Enfin, pour lancer l'application, vous devez d'abord déployer l'application struts pour que le RMI soit lancé automatiquement au déploiement (voir web.xml du projet struts). Vous pourrez ajouter des personnes dans votre population via l'interface WEB. Dans un second temps, vous pouvez lancer l'application PopulationJava (tout en ayant laissé tourner Tomcat). Cette application va juste essayer de récupérer une personne sur le serveur RMI par son ID.