[LOGO] Baba Is You

Objectif général

Le but de ce projet est de développer un jeu du type « Baba Is You » qui est un jeu où chaque niveau est une énigme qu'il nous faut résoudre en modifiant les règles de l'environnement.
Pour se faire, vous trouverez, dans chaque niveau, des phrases composées de mots (exclusivement en anglais) qui s'avèrent être des blocs déplaçable par le joueur. Chacun de ses mots va définir les propriétés de l'environnement.

Condition de rendu

Ce projet est à faire par binôme (2 personnes, ni 1 ni 3) et doit être déposé au plus tard le XX janvier 2021 à 23h59 sur e-learning.

Une soutenance « bêta » aura lieu avant les vacances de noël 2020 (la date et l'horaire vous seront précisés). Votre travail sera noté mais vous pourrez tenir compte des remarques pour améliorer votre projet jusqu'au rendu final.

Cahier des charges

1. **Les mots**
   Il existe trois catégories de mots que vous trouverez au cours de la partie: les noms, les opérateurs et les propriétés Vous y trouverez par exemple (listes non exhaustives) :
   Des **noms** tels que « Baba », « Flag », « Wall », « Water », « Skull », « Lava » ou « Rock »
   Les **opérateurs** tels que « Is » (d'autres existent tel que « On », « Has » ou « And » mais nous ne nous en occuperons pas au premier stade)
   Les **propriétés** tels que « You », « Win », « Stop », « Push », «Melt», « Hot », « Defeat » ou « Sink »

   Les mots précédents sont requis pour les premiers niveaux que vous aurez à faire, voici quelques explications quand aux propriétés de chacun de ses mots:
   - Flag, Wall, Water, Skull, Lava et Rock sont des éléments du décor
   - Is permet d'associer un nom à une propriété ou un nom à un nom

   Pour ce qui est des propriétés:
   - You, représente les éléments du décor que vous jouez
   - Win, représente les éléments de victoire, c'est à dire un des éléments du décor sur lequel vous devrez marcher pour gagner
   - Stop, représente l'élément du décor qui ne peut être traversé
   - Push, représente l'élément du décor qui peut être poussé horizontalement ou verticalement
   - Melt, représente l'élément du décor qui se détruit s'il rencontre de la chaleur (cf Hot)
   - Hot, représente l'élément du décor qui émet une chaleur
   - Defeat, représente l'élément du décor qui tue le joueur (cf You)
   - Sink, représente l'élément du décor qui se détruit au contact d'un autre élément et détruit également ce dernier

   Il vous faudra aussi ajouter 2 nouveaux mots **exclusifs** (c'est à dire que le binôme voisin n'a pas les mêmes et qu'ils existent pas dans le jeu original) à votre jeu: 1 nouveau nom et 1 nouvelle propriété
2. **Les règles changeantes**
   Vous débutez chaque niveau avec des combinaisons de ses mots-clefs déjà formées qui vous donnent les règles de l'environnement.
   Une règle est dite valide si elle peut se lire de gauche à droite OU de haut en bas, c'est à dire que ...

   ```
   |------|
   | Rock |
   |------|
   | Is   |
   |------|
   | You  |
   -------
   ```

   et

   ```
   |-----|----|-----|
   |Rock | Is | You |
   |-----|----|-----|
   ```

   ... sont deux règles valides qui disent que vous jouerez donc le ou les rochers dans le niveau (car le rocher c'est vous!).

   En revanche...

   ```
   |------|
   | You  |
   |------|
   | Is   |
   |------|
   | Rock |
   -------
   ```

   et

   ```
   |----|----|------|
   |You | Is | Rock |
   |----|----|------|
   ```

   ... ne fonctionnent pas ! Les diagonales ne sont pas non plus acceptées.

   Un mot peut être utilisé plus d'une fois si la règle ainsi formée respecte les conditions plus haut, par exemple

   ```
                    |------|
                    | Flag |
                    |------|
                    | Is   |
   |-----|----|------|
   |Rock | Is | You  |
   |-----|----|------|
   ```

   Vous jouerez donc (en simultané) les drapeaux et les rochers

   ```
          |-------|
          | Water |
   |------|-------|------|
   | Wall | Is    | Stop |
   |------|-------|------|
          | You   |
          |-------|
   ```

   Vous jouerez donc l'eau (toutes les cases d'eau), et les murs ne seront pas franchissables
   Il vous est possible d'ajouter plusieurs propriétés à un même nom, par exemple

   ```
   |------|----|------|
   | Wall | Is | You |
   |------|----|------|
   | Is   |
   |------|
   | Win  |
   |------|
   ```

   ou

   ```
          |-------|
          | Wall  |
   |------|-------|-----|
   | Wall | Is    | You |
   |------|-------|-----|
          | Win   |
          |-------|
   ```

   Avec ces deux exemples, vous jouerez donc les murs et tous les murs permettent de finir le niveau (ah oui, vous avez donc gagné, félicitations!)
   Ces associations ne se font pas qu'entre nom/opérateur/propriété car il est également possible de faire nom/opérateur/nom!

   ```
   |-----|----|------|
   |Wall | Is | Rock |
   |-----|----|------|
   ```

   Dans cet exemple-ci, nous déclarons que tous les murs de l'environnement se transforment en rochers! (Il n'y aura donc plus aucun mur)
3. **Les règles immuables**
   Il existe aussi des règles qui doivent régir chaque niveau, ces règles-là ne peuvent pas être changée.
   - « You » représentant le joueur, si aucun nom n'est associé à « You », ou qu'il n'y a plus d'élément en jeu représentant ce nom, alors la partie est perdue (si « Rock is You » mais qu'il n'y a plus de rocher en jeu... c'est fini!)
   - « Win » représentant l'élément qui déclenche la victoire, si aucun nom n'est associé à « Win », ou qu'il n'y a plus d'élément en jeu représentant ce nom, alors il n'est pas possible de finir le niveau (il vous faudra associer cette propriété à un autre nom)
   - Chaque niveau est d'une taille fixée et reste borné, ce qui empêche quelconque élément de quitter la zone de jeu
     Pensez que votre jeu doit fonctionner même sur un écran "rétina" genre 3000x2000 pixels, donc les élements graphiques ne doivent pas avoir une taille fixe !
   - Un élément qui contient du texte, les blocs, sont implicitement poussable (`Push`). Lorsqu'ils sont en ligne ou en colonne, si on pousse le premier block, cela pousse les blocs suivant
   - Un élément dont le nom n'a plus de propriété, reste figé en jeu et il peut être traversé
   - Chaque niveau peut avoir des éléments décoratifs qui n'ont aucun nom les représentants (vous pouvez avoir des fleurs sur le monde sans que celles-ci aient un bloc "Flower" de présent), dans ce cas-là ils suivent les règles ci-dessus.
4. **Les niveaux**
   Il vous faudra représenter à l'identique les 6 énigmes que vous trouverez dans la vidéo YouTube dont le lien se trouve dans la section référence ci-dessous.
   Ensuite, il vous faudra créer par vous-même **au minimum** un niveau exclusif à votre jeu qui reprendra les mots-clefs exclusifs que vous avez ajoutés précédemment.
   Pensez que votre niveau supplémentaire doit être challegeant mais que votre chargé de TP peut être un très mauvais perdant et que s'il n'a pas besoin de beaucoup réfléchir ou qu'il trouve comment résoudre votre niveau sans user vos mots additionnels cela risque de jouer en votre défaveur.
5. **Jouabilité**
   Vous débutez chaque niveau en ayant nécessairement un élément associé à « You ». Les déplacements se feront aux touches directionnelles et entraîneront le déplacement d'une case horizontalement ou verticalement en respectant les règles actives dans le niveau et la délimitation de celui-ci.
6. **Ligne de commande**
   Il est possible de paramétrer le jeu en ligne de commande à l'aide des options suivantes :
   - --levels name: nom du dossier contenant les énigmes à résoudre, auquel cas, à chaque niveau terminé on enchaîne sur le niveau suivant;
   - --level name: nom du fichier contenant l'énigme à résoudre ;
   - --execute mot1 mot2 mot3: permet d'exécuter une phrase comme "ROCK IS FLAG", il peut y avoir plusieurs --execute pour une même ligne de commande (chut, c'est pour tricher);

   Les valeurs par défaut sont respectivement : le fichier `default-level.txt` et l'élément par défaut associé à You dans le niveau choisi
7. **Affichage**
   Pour l'affichage, on vous demande d'utiliser la bibilothèque Zen 5.
   Attention à bien appeler la méthode `render` une seule fois par frame du jeu, pas plus sinon votre jeu va se transformer en sapin de noël clignottant (c'est la saison, c'est beau aussi, mais ça fait mal à la tête).
8. **Amélioration possibles**
   Attention, les autres parties du jeu devront être finies avant que vous commenciez à considérer les améliorations suivantes :
   - ajouter les sauvegardes de milieu de partie
   - ajouter un opérateur supplémentaire, par ordre de difficulté « On » , « Has » et « End »
   - ajouter les interdictions, c'est à dire que si vous avez « Rock Is Rock », vous ne pourrez plus transformer aucun rocher en autre chose

   Toute autre amélioration sera considérée négativement.

Références

Vidéo YouTube découverte - Baba Is You
Baba Is You Wiki (non officiel)
Ant Manual pour le fichier `build.xml`.
How to create an executable jar ?.
JavaDoc.
Les entrées/sorties sur fichier.

Conditions de rendu

Le projet est à rendre au plus tard le XX janvier 2021. Le format de rendu est une archive au format zip (**tout rar, tar.gz, 7z et autre ne sera pas ouvert**) contenant:

- un répertoire `src` contenant les sources du projet;
- un répertoire `docs` contenant le manuel de l'utilisateur (`user.pdf`) et le manuel qui explique votre architecture (`dev.pdf`) au format **PDF** avec une **section dédiée** aux améliorations/corrections apportées depuis la soutenance bêta;
- un répertoire `classes` **vide** dans l'archive et qui contiendra les classes **une fois compilées**;
- un répertoire `lib` contenant les libraries dont dépend l'application;
- un jar exécutable `baba.jar` qui fonctionne avec `java -jar baba.jar` et donc qui possède un fichier manifest adéquat;
- un fichier `build.xml` (écrit à la main) qui permet de :
  - compiler les sources (target compile);
  - créer le jar exécutable (target jar);
  - générer la javadoc dans `docs/doc` (target javadoc);
  - nettoyer le projet pour qu'il ne reste plus que les éléments demandés (target clean).

Cette archive Zip (attention à l'encodage) aura comme nom `Nom1_Nom2_BabaIsYou.zip`, où les noms sont ceux des membres du binôme par ordre alphabétique. L'extraction de cette archive devra créer un répertoire de nom `Nom1_Nom2_BabaIsYou` contenant tous les éléments demandés ci-dessus.

Notation

- Cas de 0 sans aucune correction (mort subite) :
  - projet **non effectué en binôme** (c'est-à-dire 2 personnes !) sans l'accord préalable de l'intervenant de TD;
  - absence à la soutenance bêta;
  - projet envoyé après la date;
  - projet non déposé sur elearning;
  - une archive qui n'a pas le bon nom;
  - fichier d'archive dont l'extraction ne produit pas un répertoire qui a le bon nom;
  - le projet utilise une ou des librairies externes autre que celles indiquées dans le sujet;
  - présence de code copié / collé du net;
  - l'utilisation de classes de `java.io` autre que `InputStream`/`OutputStream`, `Reader`/`Writer` et l'exception `IOException`;
  - l'absence de javadoc, ou javadoc pas en anglais.
- Critères de notation:
  - Rémi Forax Junior (8 ans) doit pouvoir jouer à votre jeu;
  - la propreté et la lisibilité du code auront un poids très important dans la note;
  - l'architecture que vous aurez définie (interfaces, classes ...) devra être donnée dans les documents PDF et aura également un poids très important dans la note;
  - pas de méthodes de plus de 20 lignes !;
  - pas de duplication de code, et respect des principes de programmation objet;
  - pas de variable globale;
  - pas de code inutile;
  - présence des différents rapports et, par conséquent, orthographe correcte!
  - prise en considération des remarques faites lors de la bêta pour le rendu final.