Licence 3 Informatique
Module : théorie des langages
et compilation

Turtl GUI

Fonctionnement de la sortie (interface graphique)

Information critique

La date de rendue est repoussée au jeudi 20 décembre.

1 Copie des fichiers dans le répertoire du projet

Attention : les fichiers driver.cc driver.hh main.cc vont être écrasés. Pensez à conserver une sauvegarde de vos fichiers ! !

Décompressez l'archive tortueGUI.zip dans le répertoire souhaité et copiez l'ensemble des fichiers dans le répertoire de votre projet.

Les fichiers driver.cc driver.hh et main.cc contiennent les instructions nécessaires pour lier votre parseur à l'interface graphique. Vous pouvez alors copier les fonctions de vos anciens fichiers driver.cc et driver.hh dans les nouveaux fichiers.

2 Utilisation de l'interface graphique

Pour utiliser l'interface graphique nous manipulerons un objet Jardin appelé monJardin présent dans votre driver. Un jardin possède des getters et setters permettant de modifier l'état de celui-ci. Le jardin est constitué des éléments suivants :

- Une fenêtre (définie par une largeur et une hauteur) ;
- Un vecteur de tortues (décrivant les tortues présentes dans le jardin) ;
- Un vecteur de murs (décrivant les murs présents dans le jardin) ;
- Une pile de mouvements (permettant l'exécution de l'animation) ;

Chaque case du jardin mesure 40x40 pixels ce qui correspond à la taille d'une tortue ou d'un mur. Les différentes commandes possibles sont les suivantes :

- Setters :
  - `changePosition(n, x, y)` : positionne la tortue numéro `n` aux coordonées `(x;y)` ;

- `changeOrientation(n, o)` : oriente la tortue numéro `n` à `o` degrés ;
- `changeCouleurCarapace(n, r, g, b)` : change la couleur de la carapace de la tortue numéro `n` avec la couleur RGB `(r,g,b)` avec `r`, `g` et `b` compris entre 0 et 255 ;
- `changeCouleurMotif(n, r, g, b)` : change la couleur du motif de la tortue numéro `n` avec la couleur RGB `(r,g,b)` avec `r`, `g` et `b` compris entre 0 et 255 ;
- `poserStylo(n)` : active l'écriture lors des déplacements de la tortue numéro `n` avec la couleur de la carapace ;
- `leverStylo(n)` : désactive l'écriture de la tortue numéro `n` ;
- `changeTailleJardin(w, h)` : change la taille du jardin par `w` pour la largeur et `h` pour la hauteur ;
- `nouvelleTortue()` : ajoute une nouvelle tortue au jardin ;

— Getters :

- `position(n)` : retourne la position de la tortue numéro n au format QPoint (un QPoint possède 2 attributs x et y accessibles avec les fonctions x() et y()) ;
- `orientation(n)` : retourne l'orientation de la tortue numéro `n` en degrés (float) ;
- `orientation(n)` : retourne l'orientation de la tortue numéro `n` en degrés (float) ;
- `couleurCarapace(n)` : retourne la couleur de la carapace de la tortue numéro `n` en RGB au format `QColor` (un `QColor` possède 3 attributs `red`, `blue` et `green` accessibles avec les fonctions `red()`, `blue()` et `green()`, entiers compris entre 0 et 255) ;
- `couleurMotif(n)` : retourne la couleur du motif de la tortue numéro `n` en RGB au format `QColor` (un `QColor` possède 3 attributs `red`, `blue` et `green` accessibles avec les fonctions `red()`, `blue()` et `green()`, entiers compris entre 0 et 255) ;
- `styloEstPose(n)` : retourne l'état du stylo pour la tortue numéro `n`, `true` si le stylo est posé, `false` sinon ;
- `estMur(x,y)` : retourne true s'il existe un mur aux coordonnées (x ;y), false sinon ;
- `tailleJardin()` : retourne la taille du jardin au format `QSize` (un `QSize` possède 2 attributs `width` et `height` accessibles avec les fonctions `width()` et `height()`).
- `nombreTortues()` : retourne le nombre de tortues présentes dans le jardin ;

L'ajout d'une tortue s'effectue au premier emplacement disponible dans le jardin. Si aucun emplacement n'est disponible la tortue est placée en (0,0), même s'il existe déjà un élément à cet emplacement.

Les coordonnées du jardin vont de `(0;0)` pour le coin en haut à gauche jusqu'à `(longueur-1;hauteur-1)` pour le coin en bas à droite.

2.1 Compilation des sources

Pour compiler le projet avec l'interface graphique il suffit d'exécuter les commandes suivantes :

```
qmake
make
```

Pour supprimer les sources :

```
make clean
```

Pour supprimer le projet :

```
make distclean
```

Si vous souhaitez ajouter un fichier à votre projet il est nécessaire d'ajouter ces sources au fichier `tortue.pro` aux lignes `HEADERS +=` et `SOURCES +=`

2.2 Exécution

Pour tester les différents exemples il est conseillé de fournir un jardin à construire dans un fichier texte. Ce fichier texte répond au formalisme suivant :

- La longueur du jardin est déterminée par la plus grande ligne du fichier ;
- La hauteur du jardin est déterminée par le nombre de lignes du fichier ;
- Les murs sont indiqués par une `*` ;
- Les tortues sont indiquées par un `T` ;
- Tout autre caractère correspond à un emplacement vide.

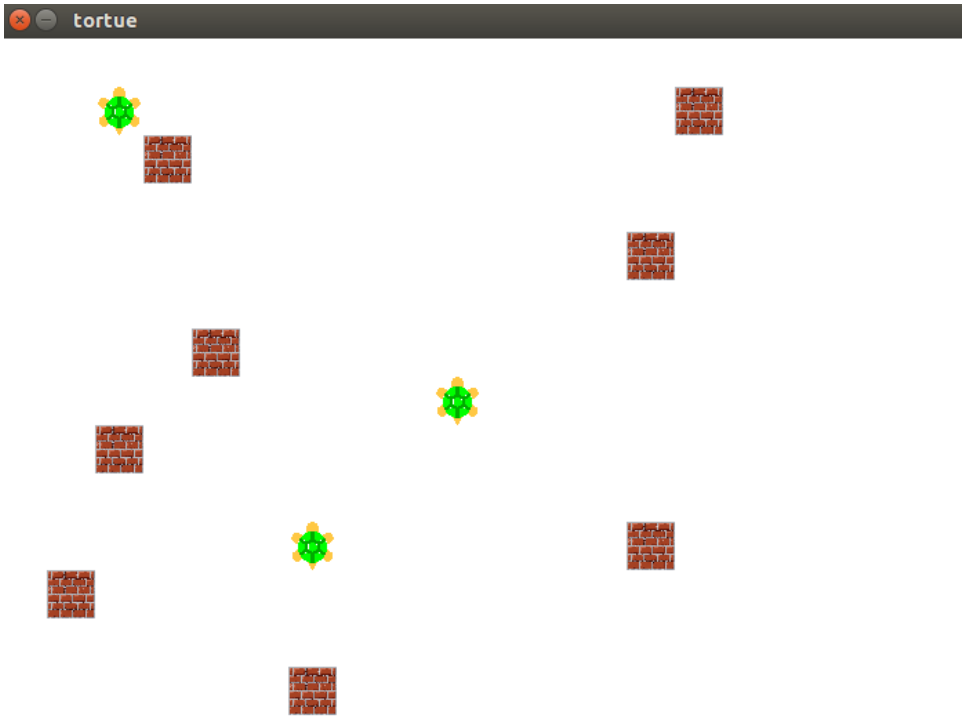
Si aucun jardin n'est fourni en entrée le jardin construit par défaut est de longueur `20` et de hauteur `15` avec une tortue en position `(0;0)`.

Les tortues insérées dans un fichier jardin sont numérotées par ordre d'apparition de haut en bas et de gauche à droite.

Voici un exemple avec le `jardin1.txt` fourni dans l'archive `tortueGUI.zip` :

```
--------------------
--T-----------*-----
---*----------------
--------------------
-------------*------
--------------------
----*---------------
---------T----------
--*-----------------
--------------------
------T------*------
-*------------------
--------------------
------*-------------
--------------------
```

Le rendu graphique est le suivant :



Exécution du programme avec un jardin en entrée et un exemple à parser :

```
./tortue jardin1.txt < 01.turtl
```

Exécution du programme sans jardin en entrée et un exemple à parser :

```
./tortue < 01.turtl
```

Lorsqu'un fichier d'exemple est fourni en ligne de commande, il n'est pas possible d'ajouter des déplacements dans le terminal.

Exécution du programme avec un jardin en entrée sans exemple à parser :

```
./tortue jardin1.txt
```

Dans ce cas nous pouvons fournir les commandes à exécuter directement dans le terminal puis exécuter les déplacements en utilisant la combinaison `[CTRL] + D`. Puis ajouter d'autres commandes par la suite.