

P1 - MVC – Cartagena (2)

Nous poursuivons la démarche de réalisation de l'interface graphique du jeu de Cartagena.

CASEVUE

Ce composant est destiné à représenter graphiquement une Case du Passage ou la Barque ou la Forteresse. Il doit pouvoir afficher l'image de son symbole et les pions qu'il porte¹. Il devra aussi pouvoir être sélectionné et désélectionné par clic de souris en émettant un `PropertyChangeEvent` (dans la mesure où sa propriété '*sélectionnable*' est à vrai).

À nouveau nous étendrons un `JPanel` pour concevoir ce composant.

Pour faciliter l'affichage, nous veillerons à ce qu'il puisse représenter ses 'murs' au travers de la méthode `void setMur(TypeMurEnum tMur, boolean b)` où `TypeMurEnum` reprend les valeurs HAUT, BAS, DROITE, GAUCHE.

Le composant devra proposer une propriété `TypeCaseEnum tCase` où `TypeCaseEnum` reprendra les valeurs PASSAGE, FORTERESSE, BARQUE, REMPLISSAGE représentant le type de case;

- x *passage* est une case du passage de la forteresse vers la barque,
- x *barque*, représente la barque,
- x *forteresse*, représente la forteresse et
- x *remplissage* permet de représenter une case vide (ce sera utile pour la représentation finale du jeu).

<i>Permet</i>	PASSAGE	FORTERESSE	BARQUE	REPLISSAGE
Murs	Oui	Non	Non	Non
Sélectionnable	Oui	Oui	Oui	Non
Pion	Oui	Oui	Oui	Non

Parmi d'autres, prévoyez les méthodes²

- `void setPirates(List<Pirate> pirates)`, permet d'afficher les pirates de la liste dans la case.

¹ Il faudra prévoir que chaque case puisse porter 24 pions. Dans la pratique les cases *passage* n'en porteront que 3 mais ça c'est le modèle qui en fixera les limites.

² En pratique il vaudrait mieux ne créer qu'une méthode `void setCase(Case cas)` mais cela complexifierait les tests.

- `void setSymbole(Symbole symbole)`

Les images utilisées seront de taille plus petite que celle de CarteVue. Pour pouvoir y accéder nous modifierons la méthode `getImageDuSymbole` en `Image getImageDuSymbole(Symbole symbole, TailleImageEnum taille)`

La classe `ConversionVersGUI` devra offrir une autre méthode statique `Color convertirCouleur(Couleur couleur)` permettant de convertir une `Couleur` en une `Color`.

TEST DU COMPOSANT CASEVUE

Pour permettre de tester la validité de votre développement, créez une `JFrame` analogue à celle présentée ci-dessous.

Les bords noirs représentent des murs. Dans l'exemple de droite il y a des murs au-dessus et au-dessous. En cliquant sur le bouton "inverser", le composant ajouterait un mur à droite.

Le menu déroulant "Gestion des pirates" doit permettre la vérification de l'affichage des pirates sur la case. Prévoyez quelques exemples pertinents (0 pirates, 1 rouge et 2 verts, ...)

