

Hashimiste

- ABID Ilyas
- AGEORGES Henzo
- BARBET Lucas
- BOULANGER Damien
- DELETANG Elie
- DUREAU Arthur
- PATAULT Kylian
- SAINTY Antoine
- TEISSIER Antoine

Introduction



Contexte du projet

Dans le cadre de la matière "Génie Logiciel 2", nous devons implémenter un jeu et des techniques d'aides.



Objectifs du projet

Implémenter le jeu choisi et des aides pour le joueur afin de l'orienter vers la victoire.



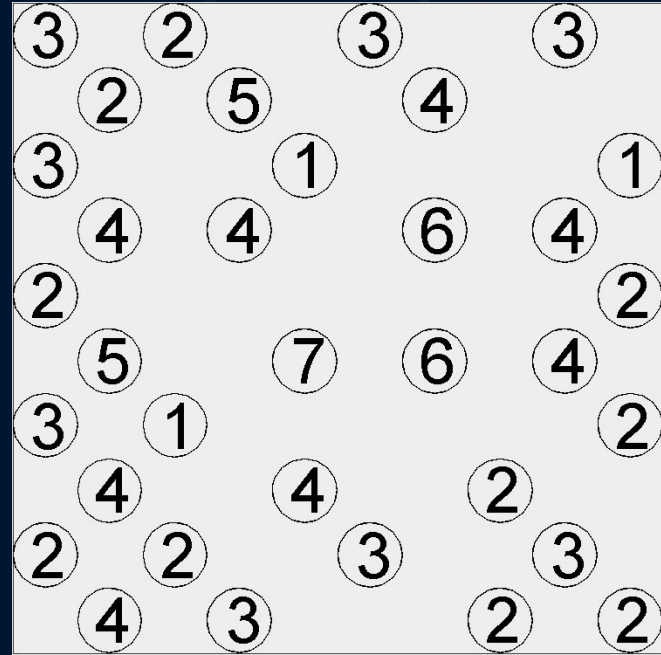
Contraintes du projet

Le jeu doit être implémenté en langage objet et doit respecter les demandes des clients.

Jeu choisi : Hashi

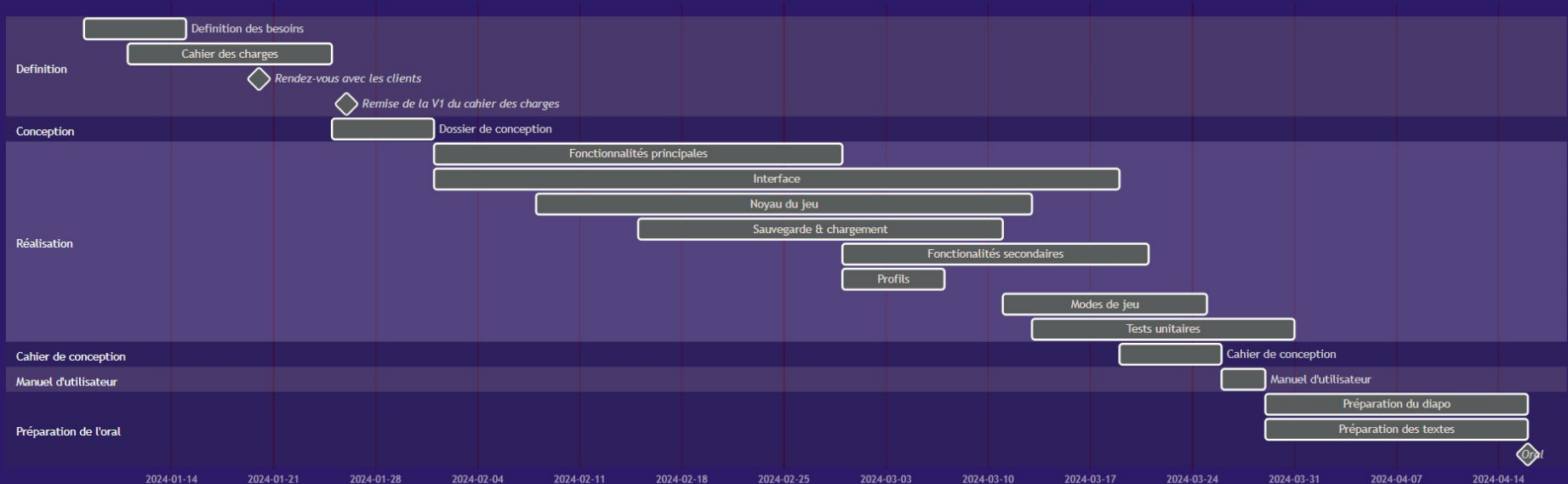
Présentation du jeu :

Le jeu du Hashi consiste à poser des ponts afin de connecter toutes les îles entre elles tout en respectant le nombre de ponts demandé pour chaque île.



Planification des tâches

Diagramme de Gantt effectif



Répartition des tâches

Interface

- ★ DUREAU Arthur
- ★ TEISSIER Antoine
- ★ AGEORGES Henzo
- ★ ABID Ilyas
- ★ DELETANG Elie

Aides et techniques

- ★ BOULANGER Damien
- * AGEORGES Henzo

Base de données

- ★ BARBET Lucas
- ★ PATAULT Kylian

Noyau du jeu

- ★ SAINTY Antoine

Fonctionnalités

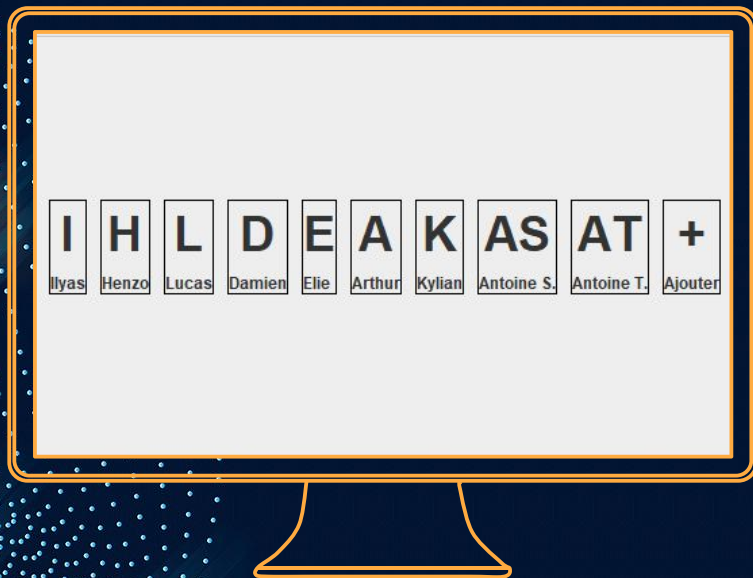
Fonctionnalités principales

- Aides en jeu
- Options en jeu
- Modes de jeu

Fonctionnalités secondaires

- Gestion de profils
 - Création / Suppression
 - Stockage de statistiques
- Graphiques
 - Thèmes et skins

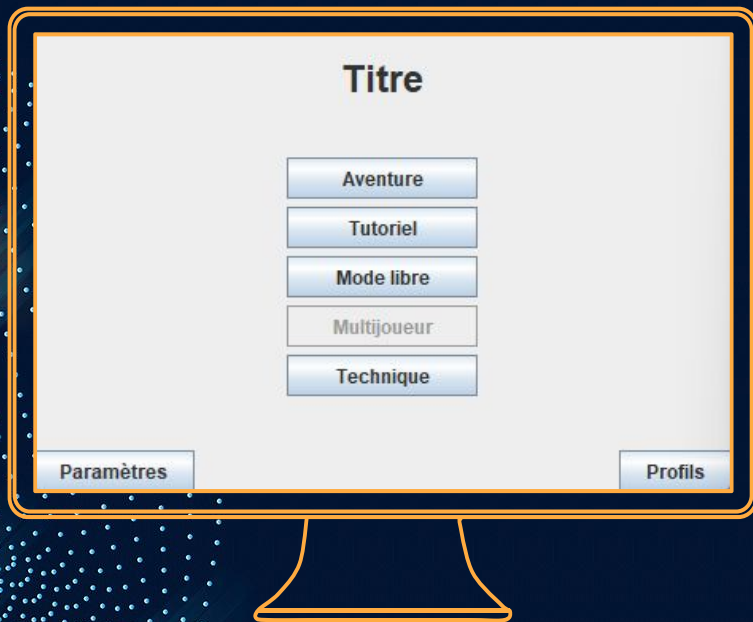
Maquettes (1/6)



Sélection de profils

- Liste des profils créés
- Chaque profil a sa sauvegarde
- Bouton “Ajouter” pour créer un nouveau profil

Maquettes (2/6)



Menu principal

- Comporte un titre
- Comporte la liste des différents modes
- Comporte un bouton "Paramètres" et un bouton "Profils" pour changer de profil

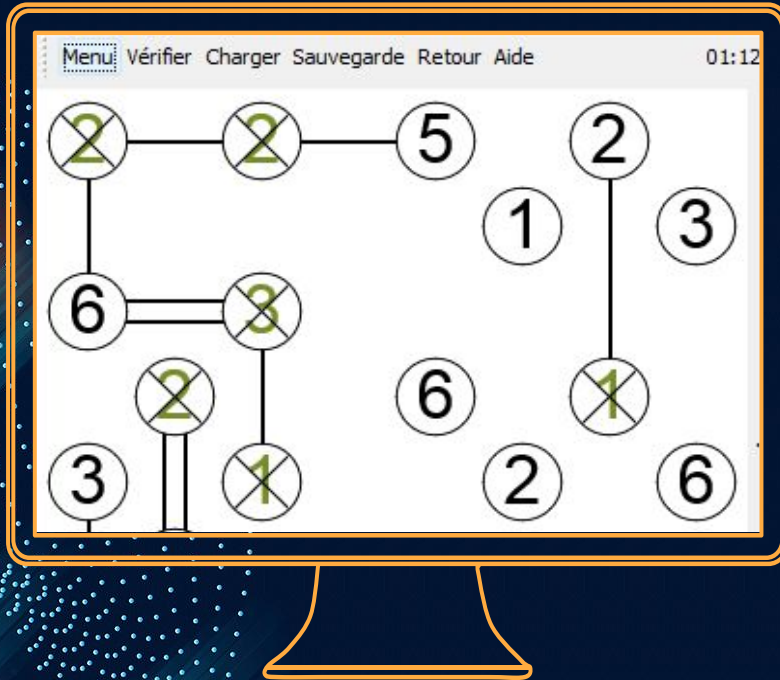
Maquettes (3/6)



Mode aventure

- 1 bouton = 1 niveau
- 1 couleur = 1 difficulté

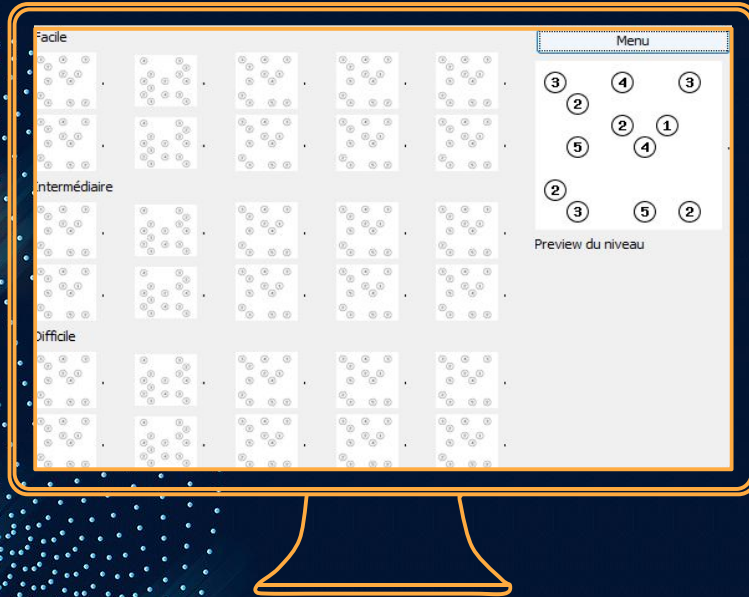
Maquettes (4/6)



Grille

- 1 rond = 1 île
- 1 trait = 1 pont
- Si une île est barrée, le nombre de ponts posé est correct
- Commandes accessibles en haut de la fenêtre

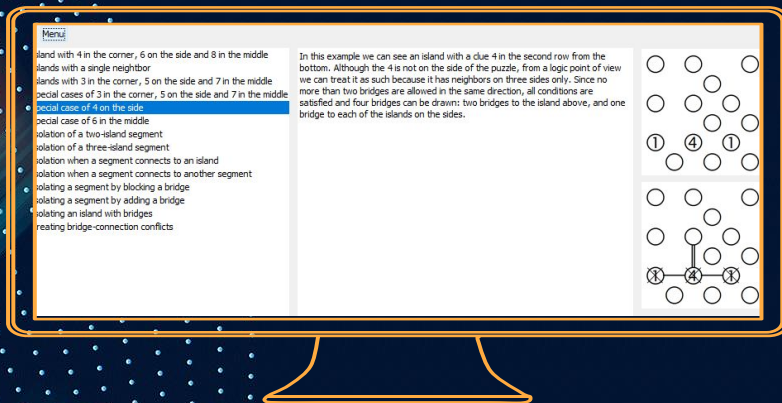
Maquettes (5/6)



Mode Libre

- Bouton menu pour revenir à l'écran principal
- Sélection de grilles au choix du joueur
- 3 niveaux de difficulté

Maquettes (6/6)



Techniques

- À gauche : la condition d'utilisation d'une technique
- Au milieu : la description de la technique sélectionnée
- À droite : un exemple illustratif de cette technique

Tests Unitaires

```
/**
 * Teste le menu
 *
 * @param menu le menu à tester
 * @param titreAttendu le titre attendu
 * @param dimensionAttendue la dimension attendue
 */
6 usages 1 MasterPNJ
protected void testerMenu(JFrame menu, String titreAttendu, Dimension dimensionAttendue) {
    // Test si les composants ne sont pas null
    assertNotNull(menu, message: "Le menu ne devrait pas être null");

    // Test si le menu est activé
    assertTrue(menu.isEnabled(), message: "Le menu devrait être activé");

    // Test si la dimension par défaut est appliquée
    assertEquals(dimensionAttendue, menu.getSize(), message: "La dimension par défaut devrait être " + dimensionAttendue);

    // Test si le titre est appliqué
    assertEquals(titreAttendu, menu.getTitle(), message: "Le titre devrait être '" + titreAttendu + "'");

    System.out.println("Le test des composants du menu réussi");
}
```

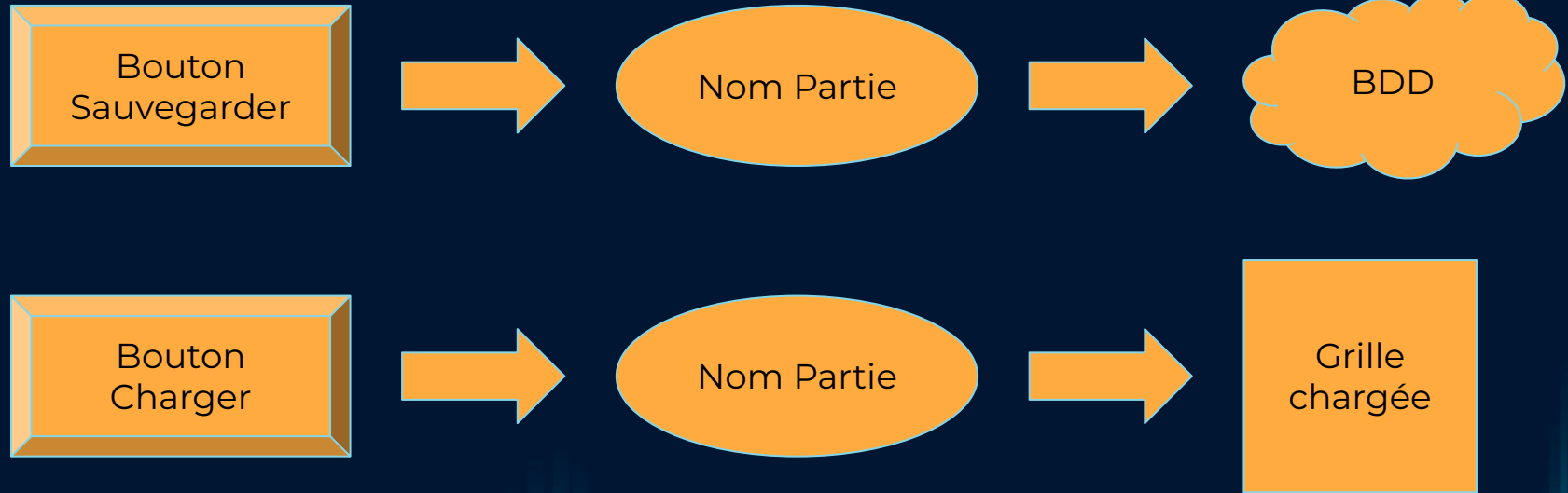

BDD

Nous avons six tables, chacune permettant le stockage ou le chargement des différents types de données.



- Grille
- Sauvegarde
- Profil
- Statistique
- Ile
- Historique

Sauvegarde / Chargement



Aide

Fonctionnement de l'aide :

- Parcours de chaque île
- Si l'île n'est pas complète :
 - Parcours des techniques
 - Si la technique s'applique :
 - On garde l'île en mémoire
- On retourne l'île avec la technique la plus simple qui s'applique



Technique 1

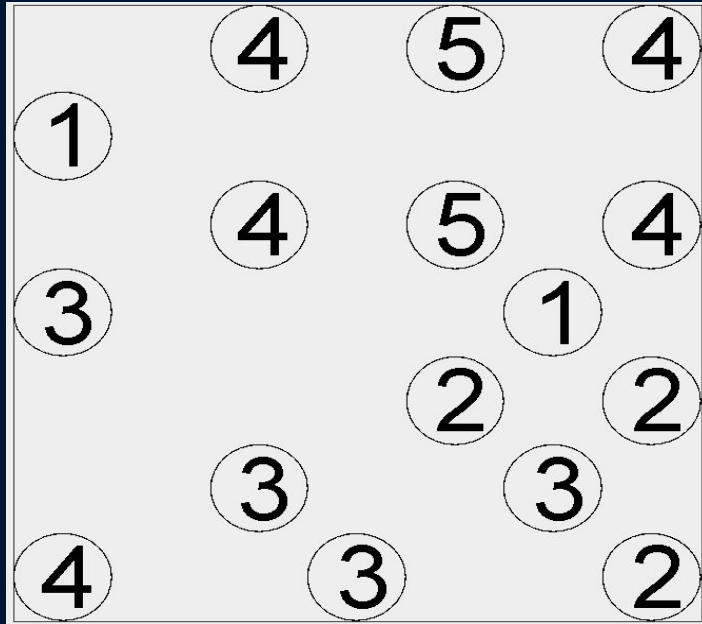
Technique 2

Technique 3

⋮

Technique n

Aide



Technique 1

Technique 2

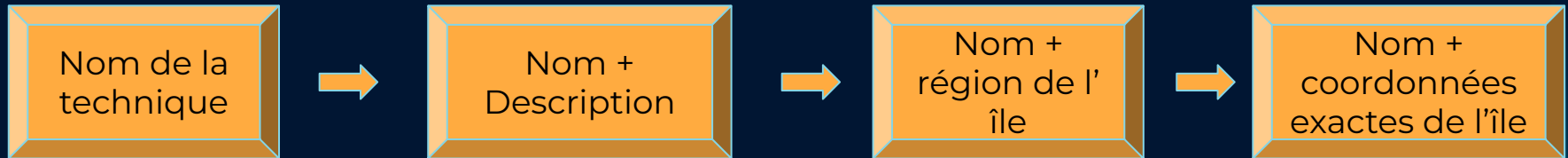
Technique 3

⋮

Technique n

Aide

Chaque clic successif sur l'aide donne des informations plus détaillées



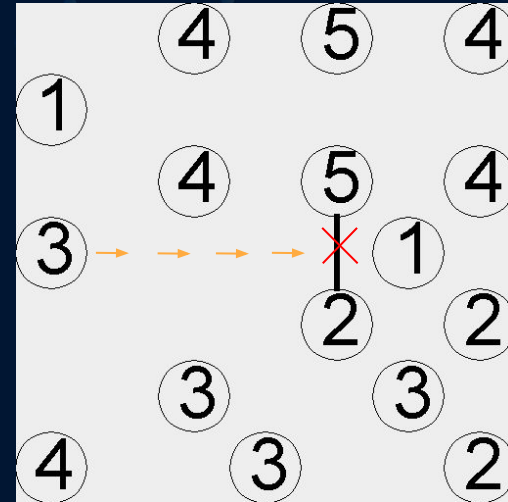
Parcours de grille

Vérification sur une ligne :

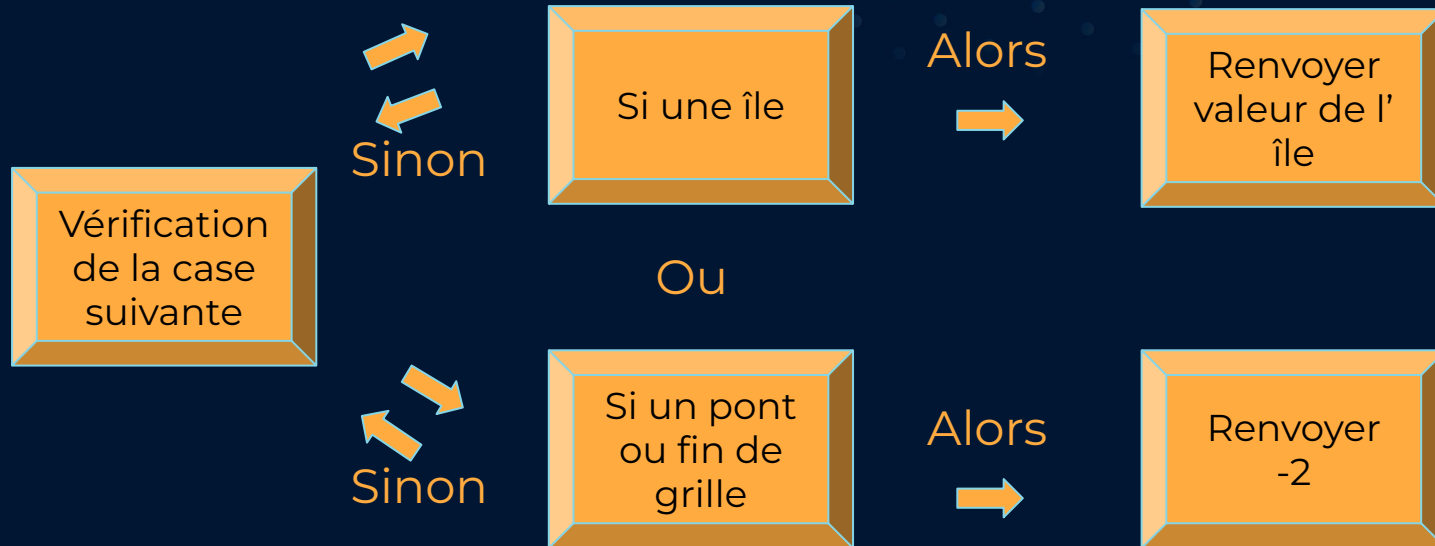
- Un pont qui bloque ?
- Pas de voisin dans la direction ?
- Valeur de l'île dans la direction ?

-> Parcours des cases dans une direction donnée

- Interface Case :
 - CaseVide
 - Pont
 - Île



Parcours de grille



Démonstration



Conclusion :

Difficultés rencontrées

- Unification des codes de chacun
 - Conflits GitHub
- Liaison entre la base de données et Java
- Mauvaise définition des besoins
- Interface graphique

