

Présentation : projet pirate

Gabernet Louis - Diallo Mariama - Grezit Gabriel - Mignotte Julie -
Smets Yoann - Tamboura Dina

présentation du jeu

- Un jeu de l'oie revisité pour 2 joueurs.
- 5 points de vie par joueurs.
- Des bombes cachées sur l'île.
- Des événements imprévisibles.

soyez le premier à trouver la barque, mais ne vous laissez pas emporter par les bizarreries de l'île...

Organisation des membres :

Louis : Chef de projet + responsable des tests fonctionnels

Dina : responsable du modèle

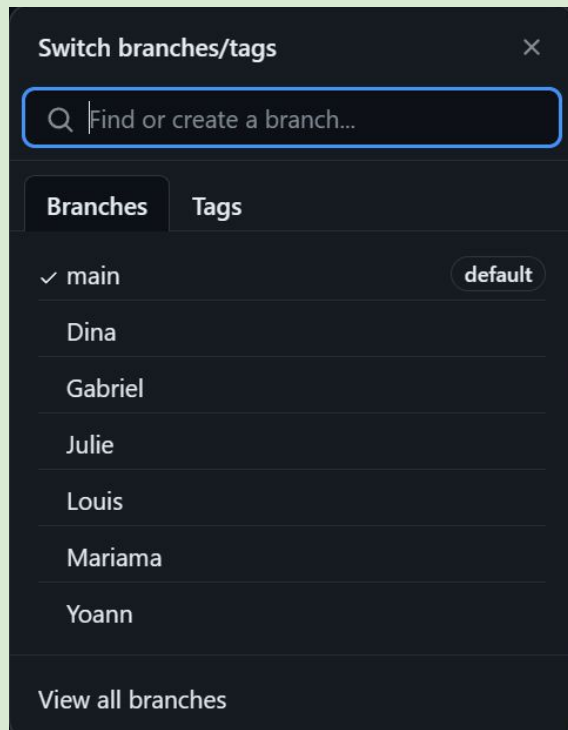
Yoann : responsable du développement du noyau

Mariama : responsable des spécifications

Gabriel : responsable du développement de l'IHM

Julie : responsable technique

GitHub

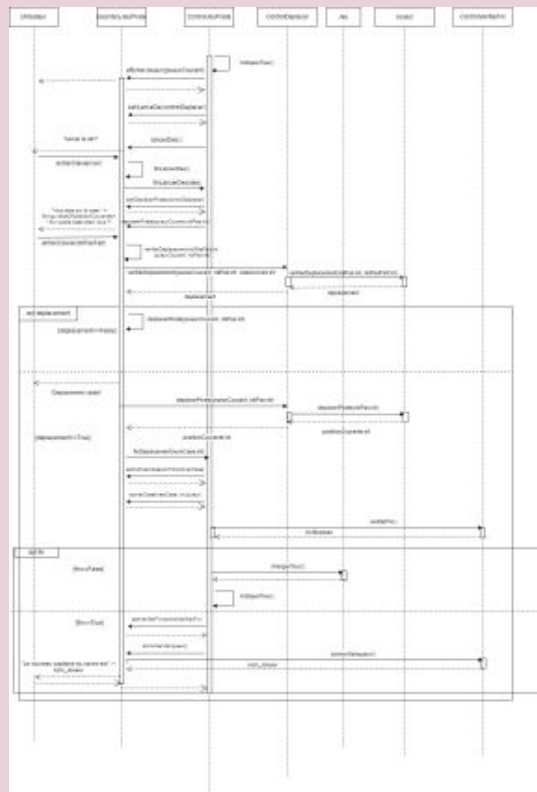


7 branches :

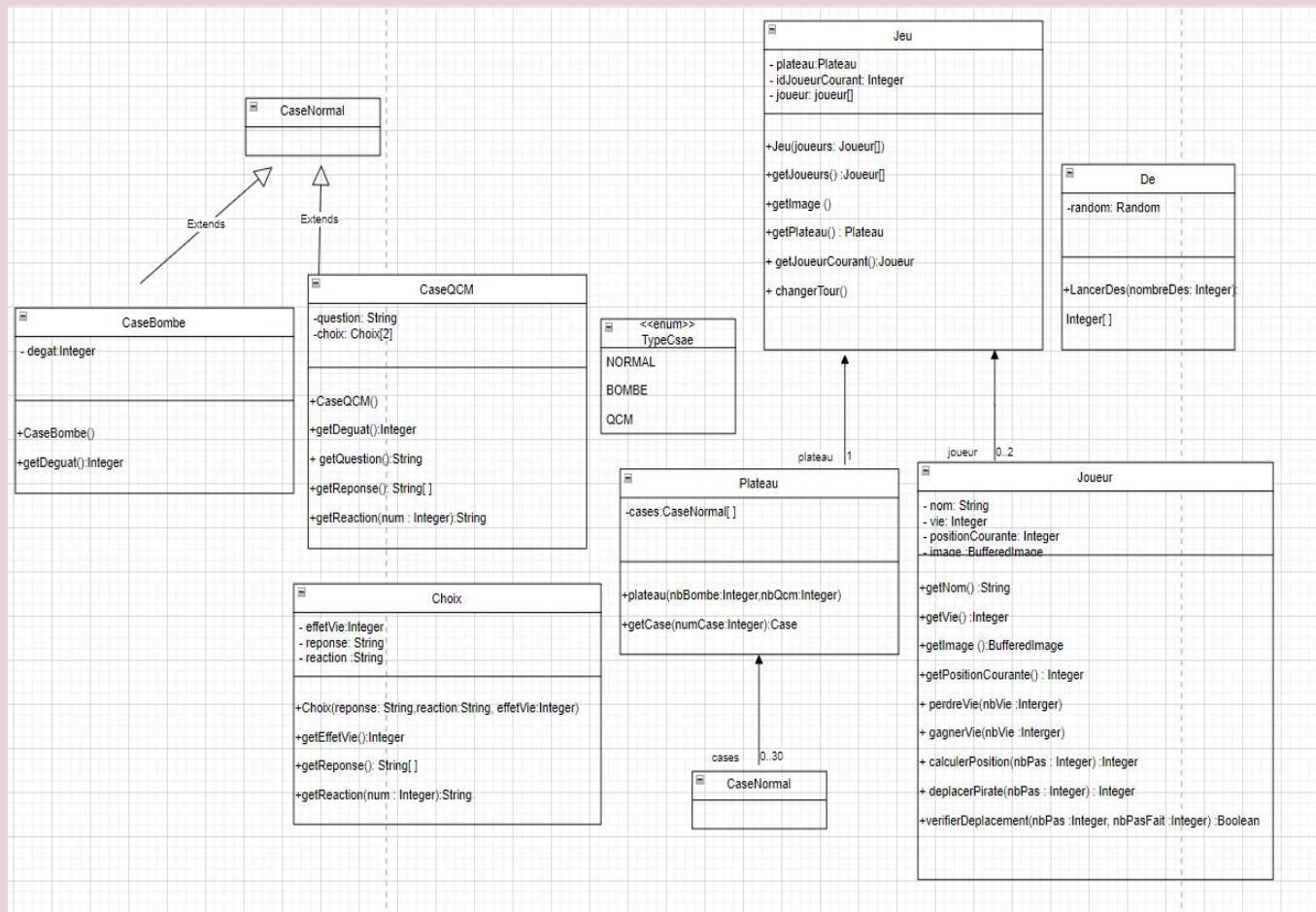
- Une branche principale
- Une branche par personne

📁 .settings	adaptateur	3 days ago
📁 bin	adaptateur	3 days ago
📁 src	debut deplacement joueur	yesterday
📁 target	debut deplacement joueur	yesterday
📄 .classpath	adaptateur	3 days ago
📄 .gitignore	adaptateur	3 days ago
📄 .project	adaptateur	3 days ago
📄 Commandes.rtf	Update Commandes.rtf	3 weeks ago
📄 README.md	Update README.md	3 weeks ago

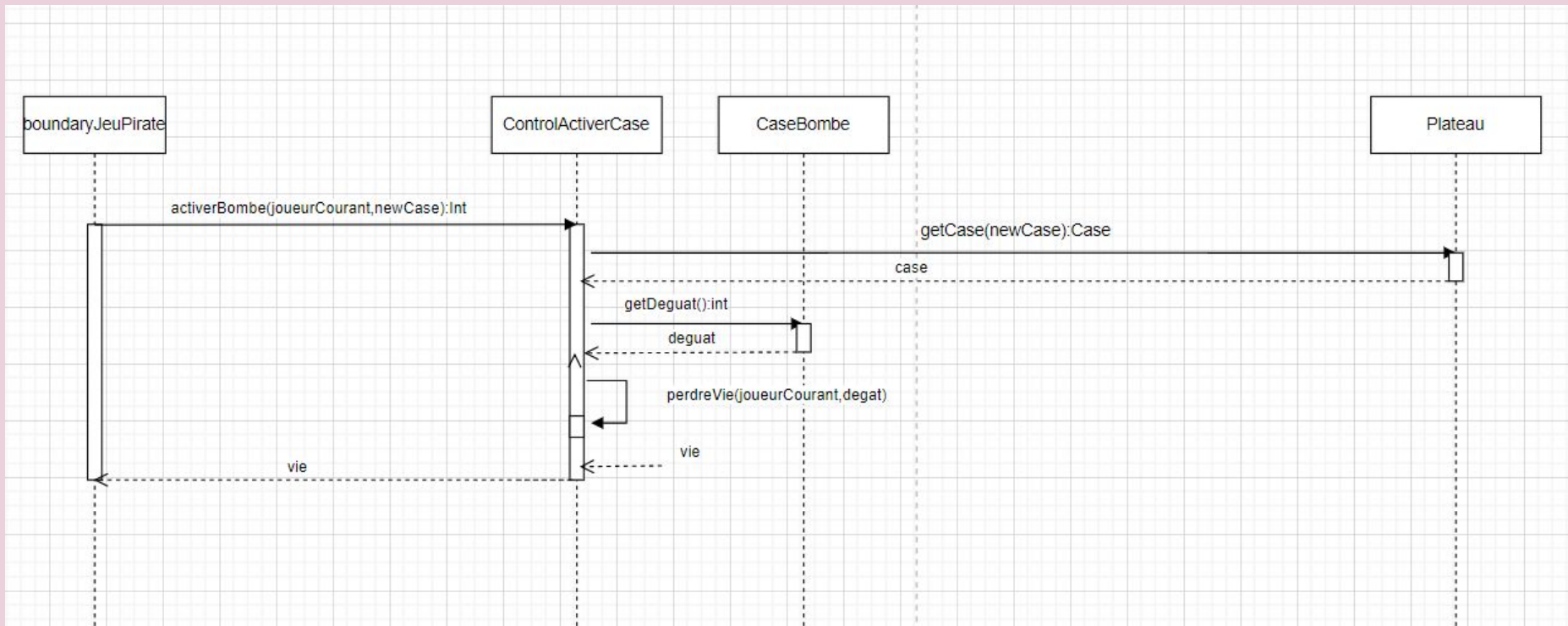
Présentation de la boucle de jeu



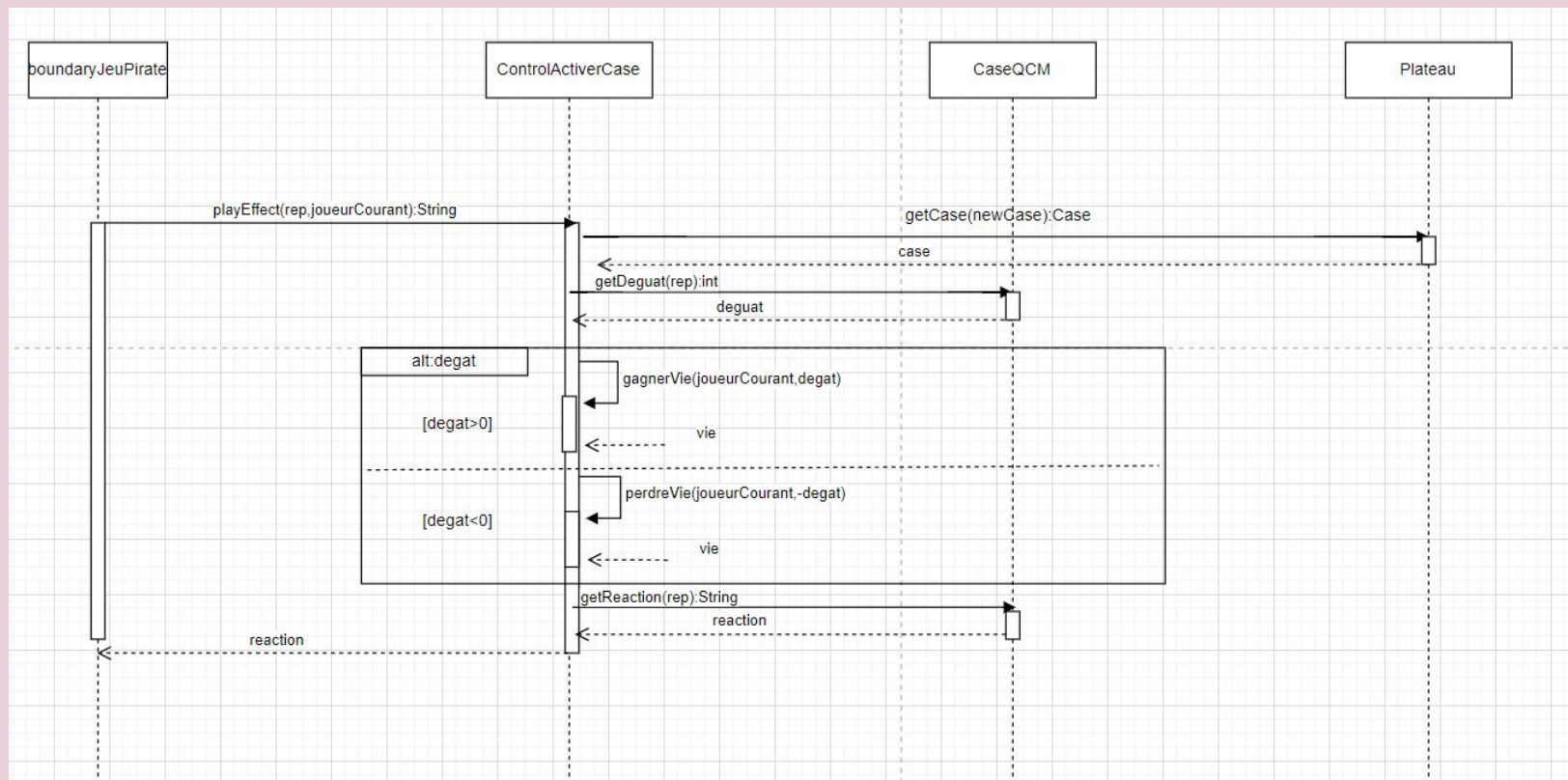
Les entités



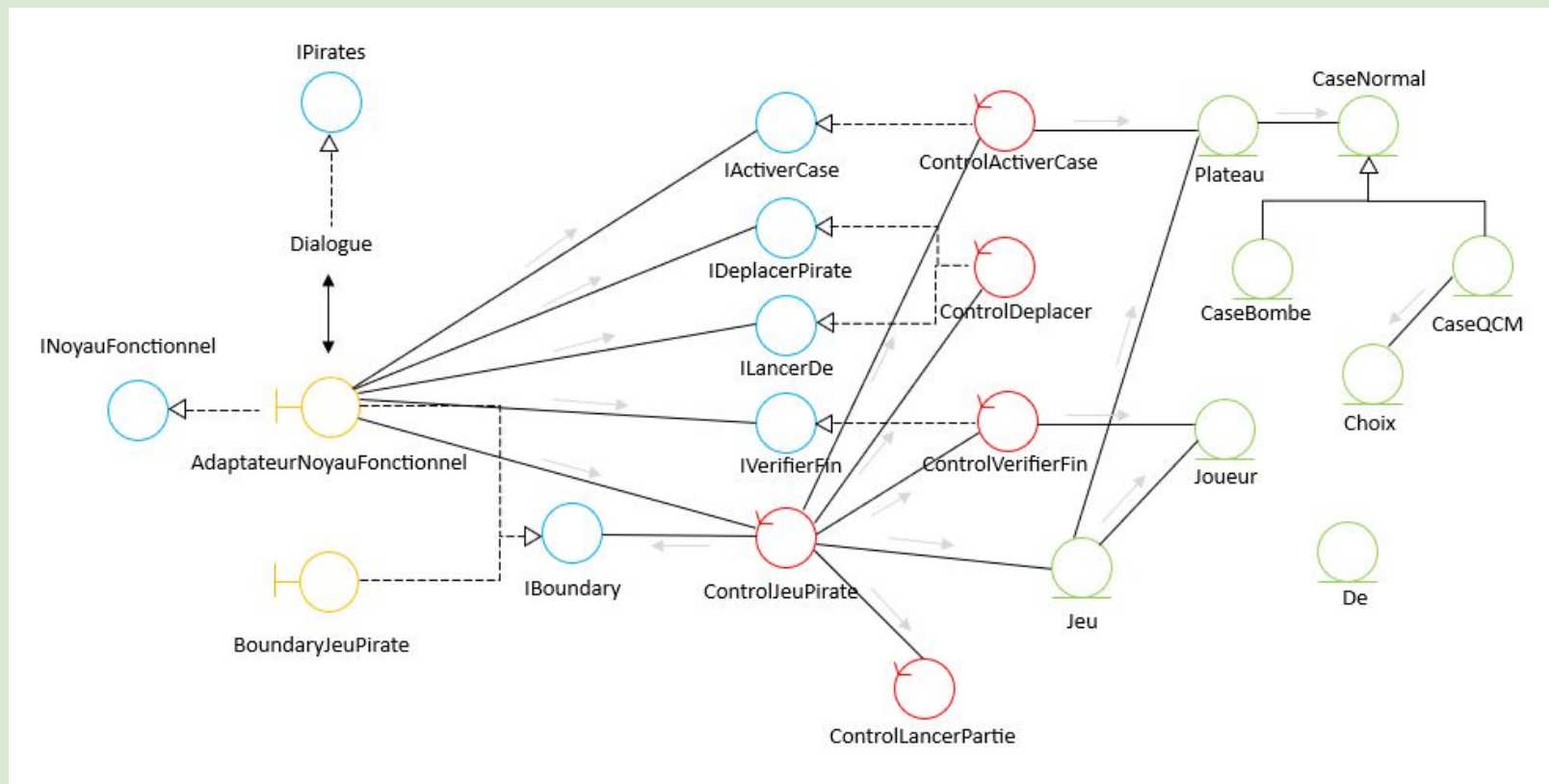
Le cas des cases spéciales : Bombe



Le cas des cases spéciales : QCM



Structure globale



Les lambdas

```
/* Create and display the form LAMBAAAA*/
java.awt.EventQueue.invokeLater(() -> {
    ControlActiverCase controlActiverCase = new ControlActiverCase();
    ControlDeplacer controlDeplacer = new ControlDeplacer();
    ControllancerPartie controllancerPartie = new ControllancerPartie();
    ControlVerifierFin controlVerifierFin = new ControlVerifierFin();

    AdaptateurNoyauFonctionnel boundary = new AdaptateurNoyauFonctionnel();
    MainFrame m = new MainFrame();
    m.setVisible(true);
    m.setAdaptateur(boundary);
    System.out.println(m);
    boundary.setiPirates(m);

    ControlJeuPirate controlJeuPirate = new ControlJeuPirate(controlActiverCase, controlDeplacer, controllancerPartie, controlVerifierFin, boundary);
    System.out.println(controlJeuPirate);
    boundary.setControlJeuPirate(controlJeuPirate);
    System.out.println(controlJeuPirate);
    controlJeuPirate.initialiserPartie();
});
```

Explication :

- aucun paramètre pour 'invokelater'
- instantiation contrôleurs
- instantiation adaptateur noyau

Les lambdas

```
@Override
public void afficherDes(int val1, int val2) {
    affichageDes1.Afficher(val1, val2);
    jButton1.setEnabled(false);
    timer = new Timer(3000, (ActionEvent e) -> {
        affichageDes1.setVisible(false);
        timer.stop();
    });
    timer.start();
}
```

Explication :

- définit l'action à effectuer lorsque le timer expire
- paramètre : `ActionEvent`
- exécution : rend l'objet 'affichageDes1' invisible et arrête le timer

La démo

- 2 joueurs
- 2 dès
- 30 cases
- Déplacement jeu de l'oie

```
C'est le tour de joueur1
Lancer le de ?
oui
Vous avez fait 3 - 3
Vous etes sur la case 1.Sur quelle case allez vous ?
Choix du joueur : 7
Deplacement valide
Rien ne se passe
```

```
C'est le tour de joueur2
Lancer le de ?
oui
Vous avez fait 1 - 5
Vous etes sur la case 1.Sur quelle case allez vous ?
Choix du joueur : 7
Deplacement valide
```

- Déplacement sur une bombe :

```
Vous etes sur la case 1.Sur quelle case allez vous ?
Choix du joueur : 4
Deplacement valide
UNE BOMBE !!!!
joueur2 a perdu 1pv. Vie actuelle : 4
Ouch ! il vous reste 4 points de vie
```

```
C'est le tour de joueur2
Lancer le de ?
oui
Vous avez fait 6 - 1
Vous etes sur la case 16.Sur quelle case allez vous ?
Choix du joueur : 23
Deplacement valide
UNE BOMBE !!!!
joueur2 a perdu 2pv. Vie actuelle : 0
Ouch ! il vous reste 0 points de vie
Fin du jeu ! Le nouveau capitaine du navire est joueur1
```

La démo

- Choix du chemin

```
Un evenement va avoir lieu..  
  
2 chemins s'offrent a vous :  
0 - aller a gauche  
1 - aller a droite  
Choix du joueur : 0  
joueur1 a perdu 0pv. Vie actuelle : 5  
Ce chemin etait tranquille
```

```
Un evenement va avoir lieu..  
  
2 chemins s'offrent a vous :  
0 - aller a gauche  
1 - aller a droite  
Choix du joueur : 1  
joueur2 a perdu 5pv. Vie actuelle : 0  
Vous vous etes perdu, vous mourrez de faim  
Fin du jeu ! Le nouveau capitaine du navire est joueur1
```

- Déplacement sur un nid :

```
Vous voyez un nid de perroquet non surveille ?  
0 - Faire cuire un oeuf  
1 - Passer son chemin  
Choix du joueur : 0  
joueur2 a perdu 2pv. Vie actuelle : 1  
On dirait bien que les parents on tout vu...
```

```
Vous voyez un nid de perroquet non surveille ?  
0 - Faire cuire un oeuf  
1 - Passer son chemin  
Choix du joueur : 1  
joueur2 a perdu 0pv. Vie actuelle : 5  
Sage decision
```

La démo

- Rencontre avec la vieille dame

```
Une vieille dame louche vous arrete et demande : pourquoi voulez-vous gagner ?  
0 - Pour le navire !  
1 - Pour etre capitaine !  
Choix du joueur : 1  
joueur1 a gagne 2pv. Vie actuelle : 4  
Vous etes un leader ne, j'adore ca !
```

```
Une vieille dame louche vous arrete et demande : pourquoi voulez-vous gagner ?  
0 - Pour le navire !  
1 - Pour etre capitaine !  
Choix du joueur : 0  
joueur1 a perdu 0pv. Vie actuelle : 5  
Vous etes d'un ennui...
```

- Découverte coffre à bouteilles

```
Un coffre est rempli de bouteilles differentes.  
0 - Boire du rhum  
1 - Boire du coca  
Choix du joueur : 1  
joueur1 a perdu 3pv. Vie actuelle : 0  
Avez vous deja oublie votre allergie ?  
Fin du jeu ! Le nouveau capitaine du navire est joueur2
```

```
Un coffre est rempli de bouteilles differentes.  
0 - Boire du rhum  
1 - Boire du coca  
Choix du joueur : 0  
joueur2 a gagne 4pv. Vie actuelle : 5  
On dirait bien que la boisson a un effet inattendu sur vous !
```

La démo

- Dépassement case d'arrivée

```
C'est le tour de joueur1
Lancer le de ?
oui
Vous avez fait 5 - 6
Vous etes sur la case 22.Sur quelle case allez vous ?
Choix du joueur : 27
Déplacement valide
Rien ne se passe
```

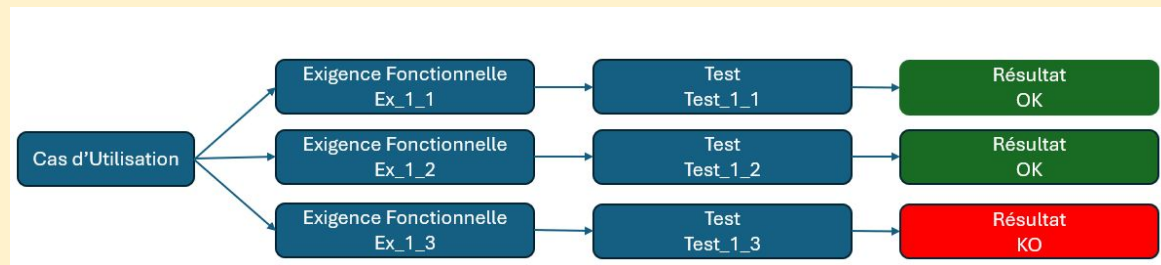
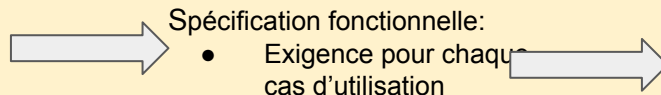
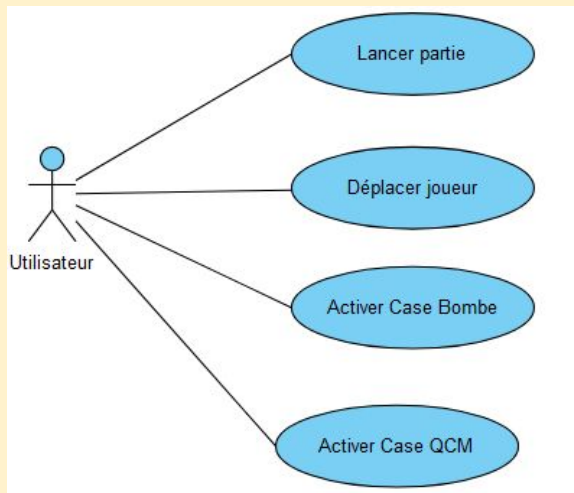
- Case courante : 22 ; Case d'arrivée : 30
 - Dépassement de 3 cases.

- Arrivée sur la case finale

```
C'est le tour de joueur1
Lancer le de ?
oui
Vous avez fait 2 - 1
Vous etes sur la case 27.Sur quelle case allez vous ?
Choix du joueur : 30
Déplacement valide
Rien ne se passe
Fin du jeu ! Le nouveau capitaine du navire est joueur1
```

Les tests fonctionnels

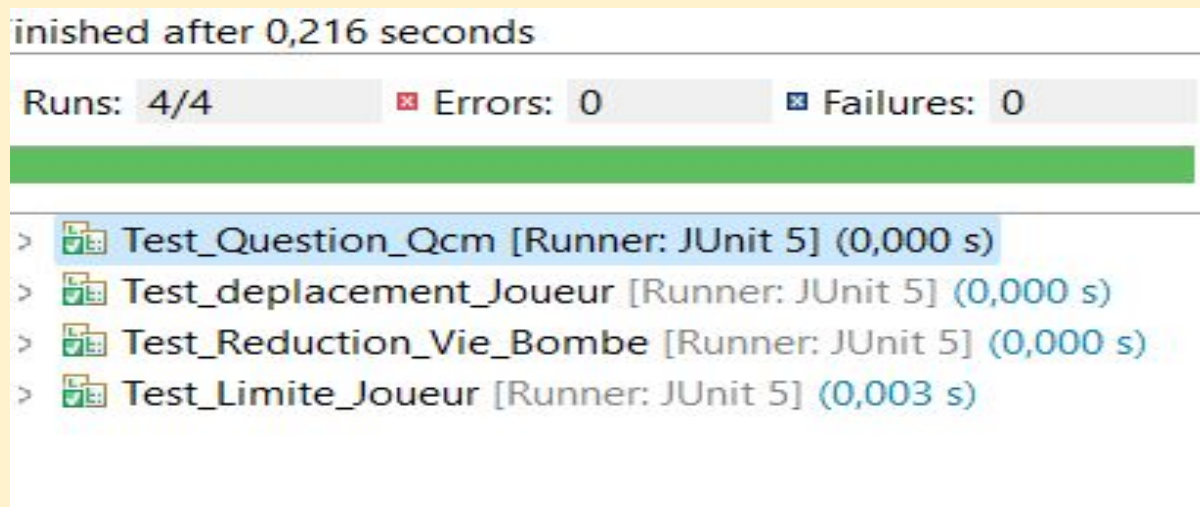
Diagramme de cas d'utilisation isolés



Les tests fonctionnels:

Un exemple de test pour chaque cas d'utilisation en ce basant sur les exigences

- **Ex_4_4** : Présenter une question à choix multiple au joueur si son pion est sur une case spéciale QCM.
- **Ex_3_2** : Autoriser le déplacement du pion du joueur selon le résultat du dé.
- **Ex_3_4** : Réduire les points de vie du joueur de manière aléatoire entre 1 et 4 points si le pion est sur une case spéciale Bombe.
- **Ex_1_1** : Implémenter une logique pour limiter le jeu à deux joueurs.



Autres tests:

Finished after 0,219 seconds

Runs: 23/23 Errors: 0 Failures: 0

TestUnitaires [Runner: JUnit 5] (0,029 s)

- testDeplacerPirateMAX() (0,000 s)
- testDeplacerPirateMIN() (0,000 s)
- testplayEffect() (0,000 s)
- testJeu() (0,000 s)
- testVerifierDeplacement() (0,000 s)
- testPerdreVieMAX() (0,001 s)
- testPlateau() (0,000 s)
- verifierFin0hp() (0,001 s)
- testVictoireJ2_case_finale() (0,001 s)
- testDeplacementPion() (0,001 s)
- testVictoireJ1_case_finale() (0,002 s)
- testQuestionQCM() (0,001 s)
- testPerdreVieSUPMAX() (0,001 s)
- testGagnerVie() (0,001 s)
- testDeplacerPirate() (0,001 s)
- testReductionPointsDeVieBombe() (0,001 s)
- testPerdreVie() (0,001 s)
- testGagnerVieSUPMAX() (0,001 s)
- verifierFin() (0,001 s)
- testVictoireJ1_0hp() (0,001 s)
- testVictoireJ2_0hp() (0,000 s)
- testLimiteDeuxJoueurs() (0,002 s)
- testLancerDes() (0,002 s)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
src	19,9 %	1 128	4 530	5 658
main/java	19,9 %	1 128	4 530	5 658
dialogue	0,0 %	0	3 862	3 862
boundary	0,0 %	0	440	440
controller	57,7 %	214	157	371
ControlJeuPirate.java	0,0 %	0	110	110
ControlActiveCase.java	71,7 %	109	43	152
ControlActiveCase	71,7 %	109	43	152
getReponses(int)	0,0 %	0	22	22
getTypeCase(int)	0,0 %	0	17	17
getQuestion(int)	91,7 %	22	2	24
playEffect(int, int, Joueur)	95,3 %	41	2	43
activerBombe(Joueur, int)	100,0 %	27	0	27
gagnerVie(Joueur, int)	100,0 %	6	0	6
perdreVie(Joueur, int)	100,0 %	6	0	6
setPlateau(Plateau)	100,0 %	4	0	4
ControlVerifierFin.java	93,7 %	59	4	63
ControlDeplacer.java	100,0 %	15	0	15
ControlLancerPartie.java	100,0 %	31	0	31
entity	84,7 %	383	69	452
TypeCase.java	0,0 %	0	34	34
CaseQCM.java	85,9 %	128	21	149
Joueur.java	94,3 %	83	5	88
Choix.java	85,7 %	18	3	21
De.java	89,3 %	25	3	28
Jeu.java	91,9 %	34	3	37
CaseBombe.java	100,0 %	16	0	16
CaseNormal.java	100,0 %	3	0	3
Plateau.java	100,0 %	76	0	76
test	99,6 %	531	2	533

Retour sur expérience



Organisation
des tâches



Compréhension de
l'architecture



Luna L'Impitoyable



Aria l'Ecarlate



William l'Ecorcheur



Callie O'Jack

START NEW GAME

