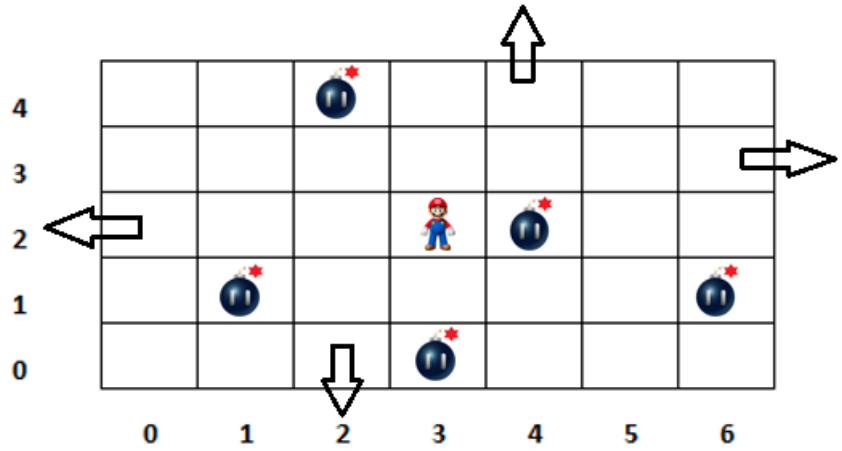
Epreuve 1ère session algorithmes et initiation à la programmation

1. Cahier de charge

Le but est créer et tester l'algorithme d'un mini jeu vidéo. Le joueur positionné initialement au centre de la grille a pour but de se déplacer et quitter la grille du jeu sans toucher une bombe. Dans le jeu de base, la grille compte 7 cases en abscisse et 5 en ordonnée. La position centrale du joueur est donc x_player = 3, y_player = 2 (3,2).



Avoir la possibilité de changer la taille de la grille de jeu et calculer la position centrale du joueur constitue une amélioration (consantes du programme).

Le nombre de bombes est de 5 (constante) mais ce nombre doit pouvoir être impérativement modifié sans modifier le code de façon significative.
Au départ du programme, la position des bombes est tirée au sort de façon aléatoire et stockée dans un tableau à deux colonnes. Pour l'exemple ci-dessus, la table `bombsPos`[][] contient :

1	1
6	1
2	4
4	2
3	0

A la base, il se pourrait que deux positions identiques soient tirées au sort, il n'y aurait alors pas réellement 5 bombes. Garantir des coordonnées distinctes constitue une amélioration. Par contre, tout programme doit garantir qu'aucune bombe ne peut avoir la coordonnée de la position centrale du joueur. L'affichage débute par la position de toutes les bombes de façon à pouvoir suivre l'évolution du jeu lors de cette phase de test.

Ensuite, le joueur peut se déplacer d'une seule case, jamais en diagonale mais uniquement vers la droite (Right), la gauche (Left), vers le haut (Up) ou vers le bas (Down) en répondant à l'invitation. `Position actuelle(x_player,y_player) déplacement RLUD/ ?` où x_player et y_player donnent la coordonnée actuelle du joueur. Le joueur ne peut donner qu'une seule lettre parmi R, L, U, D sinon un message d'erreur survient et la même question lui est reposée.

Le jeu se poursuit par des déplacements du joueur jusqu'à ce qu'il tombe sur une bombe ou qu'il arrive à sortir de la grille de jeu c'est-à-dire avoir une position x_player ou y_player en dehors des limites prévues (négative, > à 6 pour x_player, > à 4 pour y_player). Dans le premier cas, le programme affiche le message `"Bombe touchée - Vous avez perdu!",` dans le deuxième

cas, il est affiché *"Félicitations, vous avez gagné !"*. Les exemples suivants donnent des scénarios possibles de déroulement du jeu.

Exemple de déroulement « Win »	Exemple de déroulement « game over »
(6,3) (5,2) (6,4) (2,3) (6,2) ------------ Pos actu(3,2) déplacement RLUD ? R Pos actu(4,2) déplacement RLUD ? D Pos actu(4,1) déplacement RLUD ? R Pos actu(5,1) déplacement RLUD ? D Pos actu(5,0) déplacement RLUD ? L Pos actu(4,0) déplacement RLUD ? R Pos actu(5,0) déplacement RLUD ? U Pos actu(5,1) déplacement RLUD ? R Pos actu(6,1) déplacement RLUD ? R Félicitations, vous avez gagné !	(0,1) (0,4) (4,3) (5,3) (2,1) ---------- Pos actu (3,2) déplacement RLUD ? L Pos actu (2,2) déplacement RLUD ? L Pos actu (1,2) déplacement RLUD ? D Pos actu (1,1) déplacement RLUD ? L Bombe touchée - Vous avez perdu!

Créez l'application qui, au minimum (seuil de réussite), gère le jeu avec une grille de taille 7 sur 5 en positionnant le joueur initialement en (3,2). La grille comprend 5 bombes (mais ce nombre est une constante) dont les coordonnées sont tirées au sort forcément différentes de (3,2).

Les fonctionnalités suivantes (l'une et/ou l'autre) permettent d'atteindre un degré de maîtrise supplémentaire. **Cochez les fonctionnalités supplémentaires que vous envisagez !**

	Fonctionnalités	Augmentation du d. maîtrise	Choix
1	Le programme garantit que les coordonnées des bombes soient distinctes.	10%	□
2	La grille du jeu est de taille variable en ordonnée, en abscisse et la position initiale du joueur est calculée.	10%	□
3	Le message d'invitation au déplacement est RLUD? ou le ? est une saisie valide qui permet au joueur de savoir si une bombe entoure sa position actuelle (pas en diagonale et sans savoir si la bombe se trouve à gauche, au dessus, à droite ou à gauche). La réponse est alors `Bombe` ! ou `Libre` !	20%	□
4	Lié à la fonctionnalité précédente, avoir au maximum trois possibilités d'utiliser le "?" (présence bombe ?). Au-delà de 3 questions, le point d'interrogation n'est plus autorisé ni affiché dans la question. Le nombre de questions restantes se trouve noté entre parenthèse derrière le ?. Exemple : `Pos actu (3,2) déplacement RLUD?(2)`	10%	□

Exemple de déroulement fonctionnalité 3	Exemple de déroulement fonctionnalité 4
(1,0) (5,0) (2,3) (0,2) (3,1) Pos actu (3,2) déplacement RLUD?(3) L Pos actu (2,2) déplacement RLUD?(3) L Pos actu (1,2) déplacement RLUD?(3) ? Boom Pos actu (1,2) déplacement RLUD?(2) D Pos actu (1,1) déplacement RLUD?(2) L Pos actu (0,1) déplacement RLUD?(2) L Félicitations, vous avez gagné !	(1,4) (6,0) (6,4) (1,2) (5,3) Pos actu (3,2) déplacement RLUD?(3) ? Libre Pos actu (3,2) déplacement RLUD?(2) R Pos actu (4,2) déplacement RLUD?(2) ? Libre Pos actu (4,2) déplacement RLUD?(1) R Pos actu (5,2) déplacement RLUD?(1) ? Bombe ! Pos actu (5,2) déplacement RLUD ? Erreur de saisie, votre choix doit être parmi RLUD une lettre seulement Pos actu (5,2) déplacement RLUD

2. Travail à effectuer

AA1 : mettre en œuvre une stratégie cohérente de résolution du problème posé ;

Analysez le cahier de charge et dressez la liste (déclaration) des procédures et fonctions (avec leurs paramètres) que vous allez développer. **2** procédures/ fonctions (en plus du main()) sont à développer au minimum (seuil de réussite 50%).
Structurer le programme en développant des procédures/fonctions supplémentaires utiles, donne un degré de maîtrise supplémentaire à l'AA1 (10% par fonction/procédure supplémentaire qui se justifie). La fonction randInt est donnée et n'est pas comptabilisée.

AA2 : concevoir, construire et représenter l' (les) algorithme(s) correspondant(s) ;

Sur papier, et sous forme de pseudo-code, écrire l'algorithme uniquement du main et ceux qui traitent du tirage au sort des bombes, du déplacement du joueur et de la fin du jeu (Win ou Game Over) donc pas la gestion des saisies de l'utilisateur ni des fonctionnalités supplémentaires.
Le nombre de modifications (ajouts, suppressions) par rapport à cet algorithme initial ne pourra excéder 25% des instructions prévues initialement (Seuil de réussite 50%). L'algorithme devra répondre à la fonctionnalité minimum (seuil de réussite 50%). Vous avez droit à deux demandes d'assistance, au-delà de deux, le seuil de réussite pour cet AA n'est pas atteint.

Le nombre de modifications sur l'algorithme initial (0 à 15 lignes +10%), développer un code avec un minimum d'instructions par rapport à "*une solution idéale" (de -15 à +15%)*, développer les fonctionnalités optionnelles 1° (+10%), 2° (+10%), 3° (20%), 4°(10%).
Le nombre de demandes d'assistance (-10%/demande) influencent le degré de maîtrise de l'AA2 (de 50 à 100%).

AA3 : justifier la démarche algorithmique et les choix mis en œuvre ;

Pour atteindre le seuil de réussite minimum (50%) :

- les variables globales doivent être utilisées uniquement quand c'est nécessaire ou justifié,
- les choix entre procédure et fonction sont exacts et justifiés,
- les types des données sont cohérents avec les informations qu'elles représentent.

Le degré de maîtrise tient compte du fait que :

- les constantes ont été identifiées (10%),
- les structures conditionnelles donnent le moins d'instructions, d'opérations logiques possibles (20%),
- le choix des boucles est adéquat, justifié par les règles de spécificité (à mettre en commentaire du code) (20%).

AA4 : développer des programmes en respectant les spécificités du langage choisi

Avant de coder, vous devez remettre au formateur un exemplaire de vos algorithmes qui en fera une copie. Cet exemplaire fera office de référence pour évaluer le nombre de modifications que vous indiquerez, par la suite, en vert dans le pseudo code. A partir de ce moment, vous pouvez utiliser votre PC.

Codez et testez votre algorithme. Créez deux classes, l'une pour le projet complet et l'autre pour tester les fonctionnalités indépendamment. Conservez toutes les lignes de test.

Le seuil de réussite (50%) est atteint si le programme compile et réalise la fonctionnalité minimum demandée, si toutes les instructions de l'algorithme sont traduites en Java et si maximum deux demandes d'assistance sont nécessaires.

Le respect des indentations (alignements du code 20%), le choix explicite des noms de variable, fonction et procédure (20%), la présence de commentaires utiles (10%), les demandes d'assistance (-10%/demande) interviennent dans le degré de maîtrise (de 50 à 100%).
AA5: mettre en œuvre des procédures de test.
Prévoir au moins un jeu de valeurs et de résultat (scénario de test), un appel de la fonction/procédure pour chaque fonction/procédure développée (seuil de réussite à 50%) Prévoir tous les cas particuliers éventuels qui permettent de valider l'application, les fonctions/procédures dans des circonstances particulières donne un degré de maîtrise supplémentaire à l'AA5 (de 50 à 100%). Les tests des procédures et fonctiones développées doivent se trouver dans la classe de test

3. Consignes – déroulement de l'épreuve

- Bien que la grille de jeu soit un tableau, il n'est pas indispensable de devoir le déclarer de cette façon ; la position des bombes, du joueur et les limites du tableau suffisent.

- Ne pas créer de nouvelles classes d'objet (POO).
- Copier coller dans votre projet, les éventuelles méthodes se trouvrant dans Utilities et ne jamais faire appel à Utilities de façon à ce que tout le code se trouve dans votre classe.
- Remettre vos feuilles de pseudo code modifié (modifications en vert) et vos deux classes (celle de test et celle de l'application complète) en format électronique à la fin de l'épreuve.
- Nommez-les classes par votre *nom de famille* et l'autre *nom de famille_Test.*
- Ecrivez une seule instruction par ligne de code.

- Une modification dissimulée qui ne sera pas indiquée en vert par rapport au document initial sera considérée comme une tentative de fraude et entraine l'annulation de l'AA2.

- A aucun moment, vous ne pouvez communiquer avec un autre étudiant.
- Coupez votre téléphone tout au long de l'épreuve.
- Le PC n'est autorisé qu'après avoir préparé les documents du AA1, AA2 et présenté votre proposition au formateur qui en fera une copie.
- Les connections internet et réseau de l'ordinateur doivent être coupées (PC mis en mode avion, déconnecté du wifi, câble réseau débranché).
- Les seules applications qui peuvent être utilisées et ouvertes sont les éditeurs de texte, de pdf et Eclispe. Aucune autre application ne peut tourner même en arrière plan. Vous êtes susceptibles d'être contrôlés durant toute la durée de l'épreuve.

Un manquement sur ces 5 derniers points entraine l'arrêt immédiat et l'échec de l'épreuve pour tous les AA.

Signature :