

Compte rendu de Shadows of the Knight - Episode 1

Lilian Hiault

02 mai 2019

Table des matières

1	Les structures requises	1
1.1	La zone de recherche	2
1.2	La position de Batman	2
2	Actualiser la zone de recherche	2
2.1	Modifier la zone de recherche	2
2.2	Position de la bombe	2
3	Déplacer Batman pour trouver la bombe	3
3.1	Déplacer Batman	3
3.2	Trouver la bombe	4

Introduction

« Shadows of the Knight - Episode 1 » est un problème posé sur CodinGame <https://www.codingame.com/training/medium/shadows-of-the-knight-episode-1>. Une bombe a été posée dans un immeuble et Batman doit l'atteindre et la désamorcer avec qu'elle n'explose. Il saute de fenêtres en fenêtres en ne dispose que d'un détecteur qui lui indique la position relative de la bombe.

Pour trouver la bombe en un nombre d'essais limité on applique une recherche dichotomique.

1 Les structures requises

Pour répondre à ce problème j'ai décidé de mettre à jour une zone de recherche dans laquelle la bombe se trouve à chaque itération de déplacer Batman en son centre. Pour cela j'ai créé deux structures.

1.1 La zone de recherche

La première correspond à la zone de recherche. chaque nombre correspond au numéro de ligne ou de colonne qui délimite le secteur.

```
typedef struct zoneDeRecherche
{
    int gauche;
    int droite;
    int haut;
    int bas;
} zone;
```

1.2 La position de Batman

La seconde correspond à la position de Batman, c'est elle qui va se retrouver au centre de la zone de recherche à chaque itération.

```
typedef struct position
{
    int x;
    int y;
} position;
```

2 Actualiser la zone de recherche

2.1 Modifier la zone de recherche

Une fois la direction de la bombe obtenue, il faut changer la taille de la zone de recherche. Cette fonction permet la changer selon les nouveau paramètres reçus.

```
void initZoneRecherche(zone * pSecteurRech, position batman, int nvGauche,
    int nvDroite, int nvHaut, int nvBas)
{
    pSecteurRech->gauche = nvGauche;
    pSecteurRech->droite = nvDroite;
    pSecteurRech->haut = nvHaut;
    pSecteurRech->bas = nvBas;
}
```

2.2 Position de la bombe

Lorsqu'on reçoit la direction de la bombe il faut la prendre en compte pour changer la taille de la zone de recherche. Par exemple si elle se trouve au-dessus à droite il faut éliminer toutes les colonnes à gauche de Batman et toutes les ligne en dessous. De plus la bombe ne peut pas se trouver sur la ligne et colonne où se trouve le personnage. En reprennant cela avec tous les différents cas possible on modifie la zone grâce à la fonction `initZoneRecherche`.

```

void changerZoneRecherche(zone * pSecteurRech, char * bombeDir,
    position batman)
{
    if(strcmp(bombeDir, "U") == 0) {
        initZoneRecherche(pSecteurRech, batman, batman.x, batman.x,
            pSecteurRech->haut, batman.y-1);
    }
    else if (strcmp(bombeDir, "UR") == 0) {
        initZoneRecherche(pSecteurRech, batman, batman.x+1, pSecteurRech->droite,
            pSecteurRech->haut, batman.y-1);
    }
    else if (strcmp(bombeDir, "R") == 0) {
        initZoneRecherche(pSecteurRech, batman, batman.x+1, pSecteurRech->droite,
            batman.y, batman.y);
    }
    else if (strcmp(bombeDir, "DR") == 0) {
        initZoneRecherche(pSecteurRech, batman, batman.x+1, pSecteurRech->droite,
            batman.y+1, pSecteurRech->bas);
    }
    else if (strcmp(bombeDir, "D") == 0) {
        initZoneRecherche(pSecteurRech, batman, batman.x, batman.x, batman.y+1,
            pSecteurRech->bas);
    }
    else if (strcmp(bombeDir, "DL") == 0) {
        initZoneRecherche(pSecteurRech, batman, pSecteurRech->gauche, batman.x-1,
            pSecteurRech->bas, batman.y+1);
    }
    else if (strcmp(bombeDir, "L") == 0) {
        initZoneRecherche(pSecteurRech, batman, pSecteurRech->gauche, batman.x-1,
            batman.y, batman.y);
    }
    else if (strcmp(bombeDir, "UL") == 0) {
        initZoneRecherche(pSecteurRech, batman, pSecteurRech->gauche, batman.x-1,
            pSecteurRech->haut, batman.y-1);
    }
}

```

3 Déplacer Batman pour trouver la bombe

3.1 Déplacer Batman

Pour déplacer le personnage, je modifie les paramètres de la structure qui correspondent à sa position. Puisqu'on cherche à optimiser les déplacements par une recherche dichotomique, on place Batman au centre de la zone de recherche actuelle. Son abscisse est donc égale à l'indice de l'extrémité gauche plus celui de l'extrémité droite de la zone divisée par deux. On fait de même pour son ordonnée en calculant par rapport aux extrémités hautes et basses de la zone.

```

void deplaceBatman(zone secteurRech, position *pBatman)
{
    pBatman->x = (secteurRech.gauche + secteurRech.droite)/2;
    pBatman->y = (secteurRech.haut + secteurRech.bas)/2;
}

```

3.2 Trouver la bombe

Une fois qu'on a récupéré les données (largeur, hauteur, nombre de tours) on initialise la position de Batman grâce à la structure qui correspond. Idem pour la zone de recherche. Lorsqu'on rentre dans la boucle du jeu, à chaque itération on reçoit la direction de la bombe par rapport au super hero. On utilise alors la fonction permettant de changer la taille de la zone de recherche en fonction de l'emplacement de la bombe. Lorsque cette zone est actualisée, le personnage se déplace au milieu. On affiche alors ses coordonnées.

```

int main()
{
    int largeur;
    int hauteur;
    scanf("%d%d", &largeur, &hauteur);
    int nbTours;
    scanf("%d", &nbTours);
    position batman;
    position *pBatman = &batman;
    scanf("%d%d", &batman.x, &batman.y);
    zone secteurRech = {0, largeur-1, 0, hauteur-1};
    zone *pSecteurRech = &secteurRech;

    while (1) {
        char bombeDir[4];
        scanf("%s", bombeDir);
        changerZoneRecherche(pSecteurRech, bombeDir, batman);
        deplaceBatman(secteurRech, pBatman);
        printf("%d %d\n", batman.x, batman.y);
    }
    return 0;
}

```