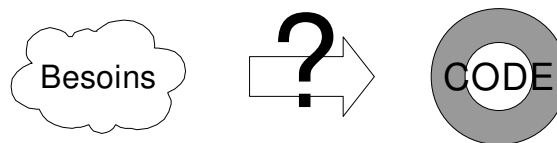


## Objectifs

Cela revient à poser la problématique : quelle démarche pour passer des besoins au code ?



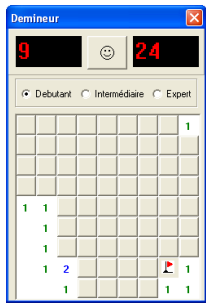
## Préambule

Les besoins sont les conditions auxquelles un système et plus généralement un projet doivent satisfaire.

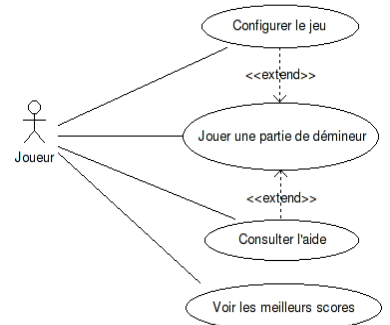
## Méthode

Il n'y a aucune obligation à utiliser tous les diagrammes de la norme UML dans un processus de développement. Justement, le processus de développement va préciser les diagrammes « utiles ».

Les artefacts doivent s'utiliser conjointement afin d'éviter les incohérences et de s'enrichir mutuellement afin d'obtenir une « bonne analyse » du problème.



Maquette



Cas d'utilisations

Modélisation des exigences  
Analyse

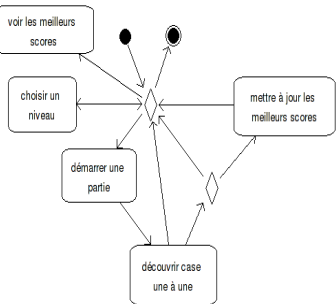


Diagramme d'activités de navigation

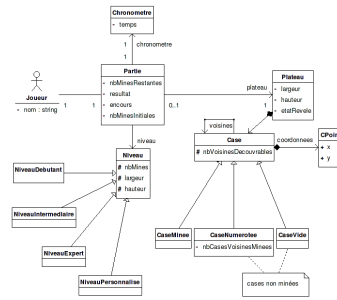


Diagramme de classes d'analyse

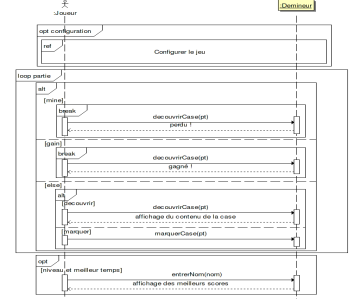


Diagramme de séquence système

Conception

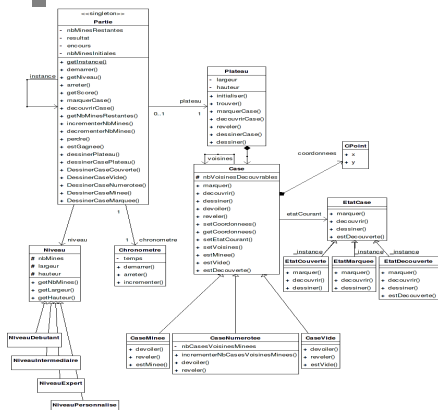


Diagramme de classes de conception

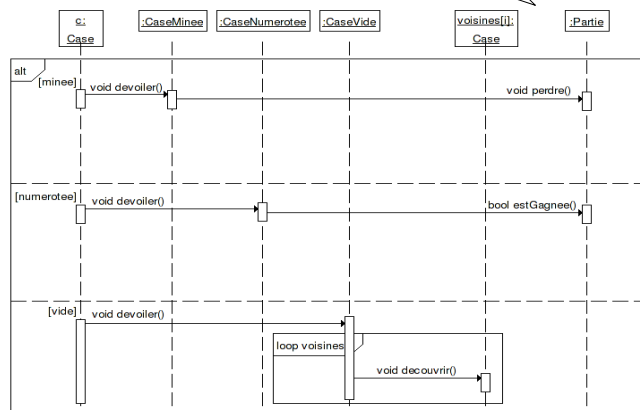


Diagramme de séquence de conception

Implémentation

```

#include <string>
#include <vector>
#include <list>
#include <map>
#include <algorithm>
using namespace std;
typedef map<CPoint, Case*, CPointInfo> Map;

class CPoint;
class CPointLess;
class Case;

class Plateau
{
private:
    int largeur;
    int hauteur;
    Map plateau;
    bool etatRevele;
    bool donnerCoordonneeCaseVoisine(int direction, CPoint* coordonneeCase);

public:
    Plateau(int nbLines, int largeur, int hauteur);
    ~Plateau();
    void initialiser(int nbMines);
    Case* trouver(CPoint pt);
    void marquerCase(CPoint pt);
    void decouvrirCase(CPoint pt);
    void revele();
    void dessinerCase(CPoint pt);
    void dessiner();
    bool estRevele();
};
    
```

Code