

C.9

Passer à la conception détaillée

Objectif

En utilisant le modèle d'analyse, obtenir les opérations liées aux classes du modèle.

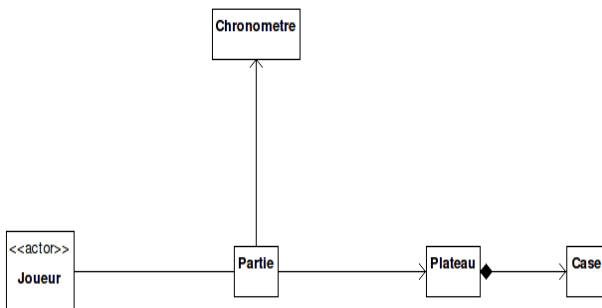
Préambule

Le travail d'analyse a permis de définir « quoi faire » et certains diagrammes ont été établis. La conception, qui est la recherche d'une solution à « comment faire », va permettre d'enrichir et d'affiner ces diagrammes.

C'est le modèle objet qui sert de cadre principal à la conception. Le modèle objet issu de l'analyse doit être complété en trouvant les opérations.

Exemple

Après avoir défini un modèle de domaine, l'analyse a permis l'identification des classes, des attributs et des associations, illustré par un diagramme de classes d'analyse.



Modèle du domaine

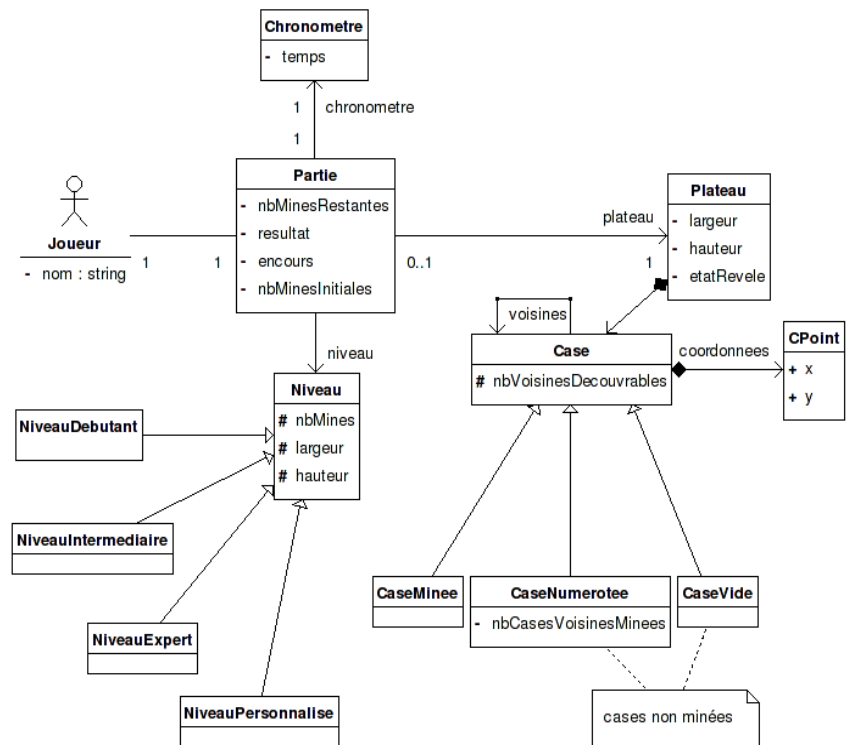
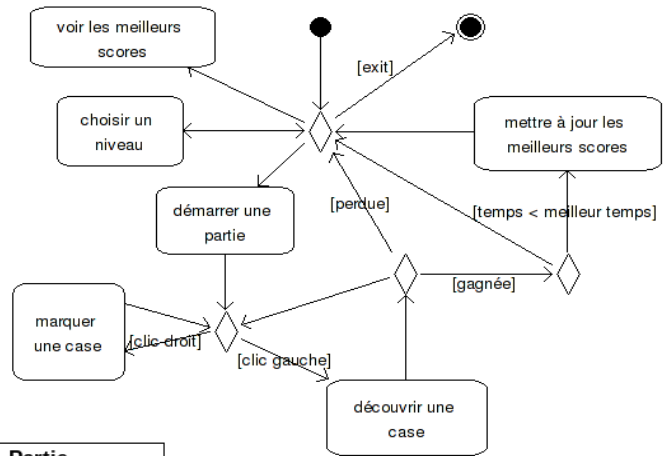
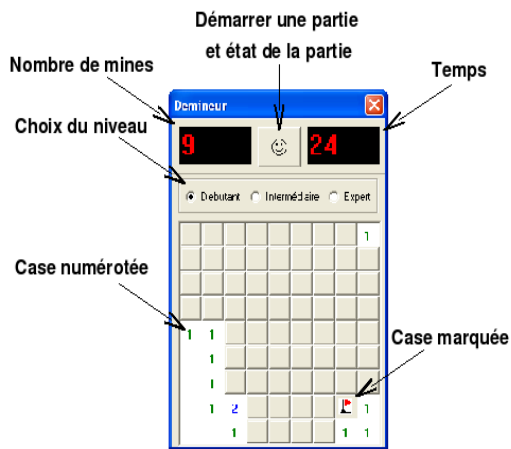


Diagramme de classes d'analyse

Méthode

On doit convertir les actions et les activités du modèle dynamique en opérations liées aux classes du modèle objet.



Partie	
-	nbMinesRestantes
-	resultat
-	encours
-	nbMinesInitiales
#	Partie()
+	~Partie()
+	démarrer()
+	arreter()
+	marquerCase()
+	decouvrirCase()
+	incrementerNbMines()
+	decrementerNbMines()

Méthode

Dans un diagramme d'états, chaque transition est une modification de l'état d'un objet et est représentée par une opération sur l'objet.

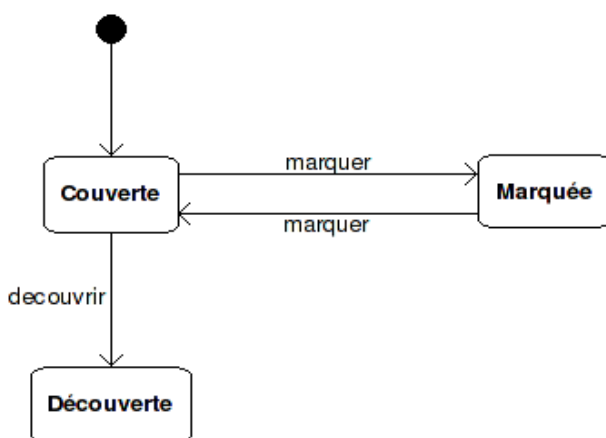


Diagramme d'états de la classe Case

Case
+ marquer()
+ decouvrir()



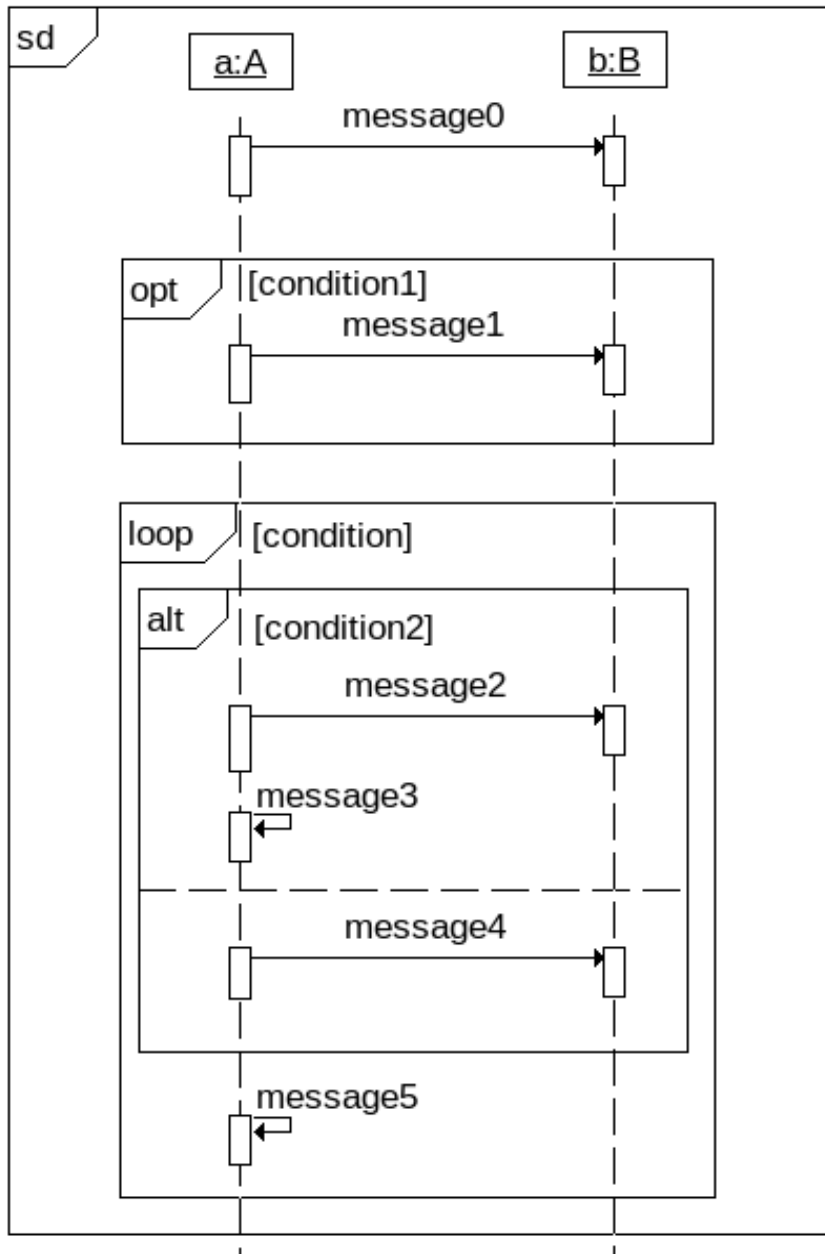
Case
+ marquer()
+ decouvrir()
+ devoiler()
+ estMinee()
+ estVide()
+ estDecouverte()

Remarque

Une opération peut être décomposée en appels à des opérations plus simples jusqu'à ce que les opérations soient suffisamment simples à implémenter.

Chaque opération système va donner lieu à un diagramme de séquence. Les diagrammes de séquence vont permettre d'élaborer le diagramme de classes de conception en ajoutant principalement les opérations.

Un message ne peut être reçu par un objet que si sa classe a déclaré l'opération publique correspondante.



Pseudo-code :

```

b.message0()
SI condition1
ALORS b.message1()
FSI
TANT QUE condition
FAIRE
    SI condition2
    ALORS b.message2()
        message3()
    SINON b.message4()
    FIN SI
    message5()
FIN TANT QUE
    
```

Code :

```

b.message0();
if(condition1)
{
    b.message1();
}

while(condition)
{
    if(condition2)
    {
        b.message2();
        message3();
    }
    else
    {
        b.message4();
    }
    message5();
}
    
```

