

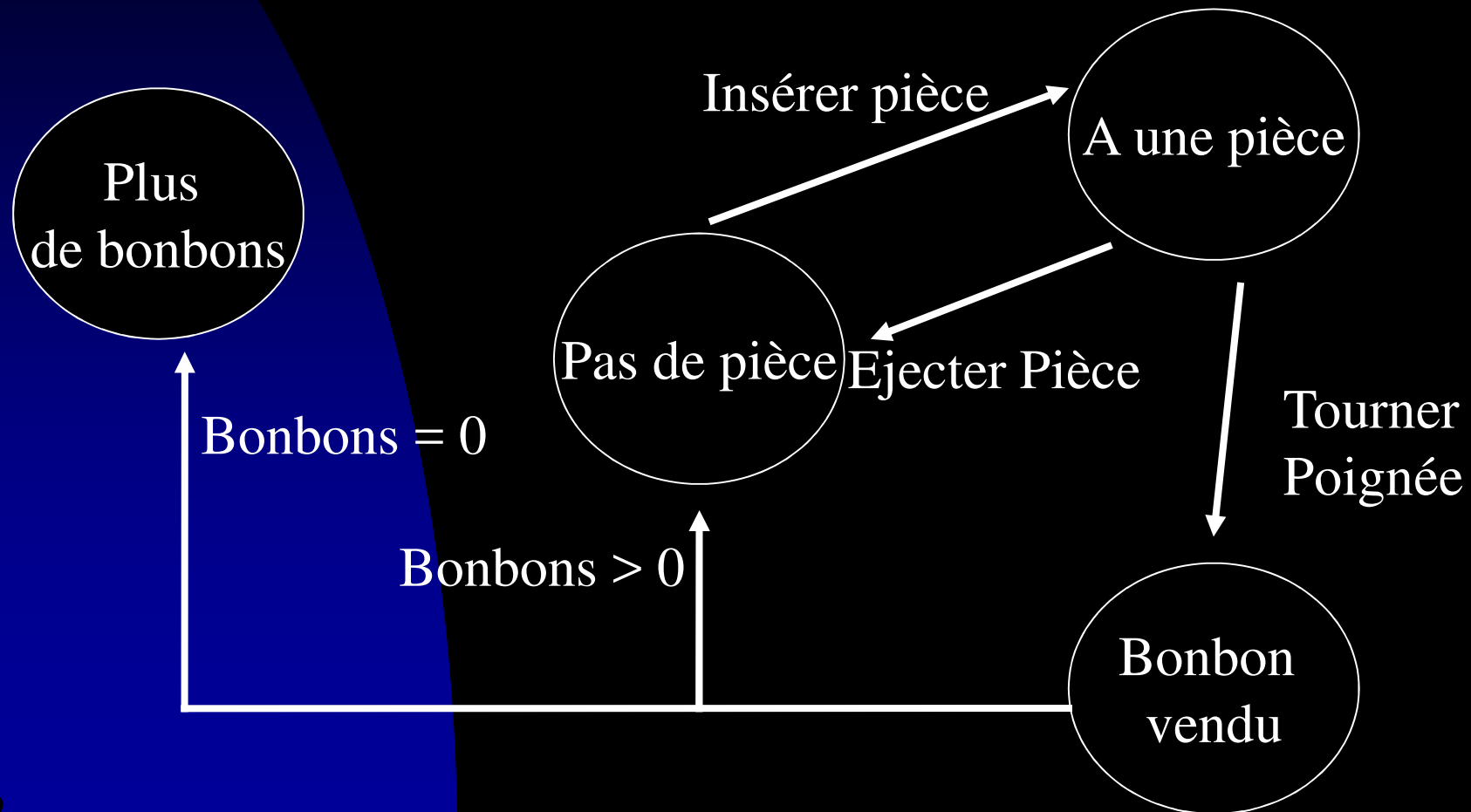
Cliquez pour ajouter un texte

LE PATTERN ETAT (STATE)

Ludovic Liétard

Pattern Etat : un exemple

- Imaginons l'application d'un distributeur de bonbons :



Pattern Etat : un exemple

- Il y a quatre états pour le distributeur
- On peut penser à modéliser l'état courant par un entier, variable d'instance du distributeur
- À chaque événement, on peut penser à écrire une méthode qui teste l'état courant

Pattern Etat : un exemple

```
class Distributeur {  
  
    final static int EPUISE = 0;  
    final static int SANS_PIECE = 1;  
    final static int A_PIECE = 2;  
    final static int VENDU = 3;  
  
    int etat;                // VI initialisée à SANS_PIECE  
    int nombre;              // VI nombre de bonbons  
  
    public Distributeur() {  
        etat = SANS_PIECE;  
        nombre = 100;  
    }  
}
```

Pattern Etat : un exemple

```
public void insererPiece() {  
    if (etat == A_PIECE) {  
        System.out.println(" Vous ne pouvez rajouter des pieces ");  
    }else if (etat == SANS_PIECE) {  
        etat = A_PIECE;  
        System.out.println(" Vous avez inséré une pièce ");  
    }else if (etat == EPUISE) {  
        System.out.println(" Plus de bonbon : pièce rejetée ");  
    }  
}
```

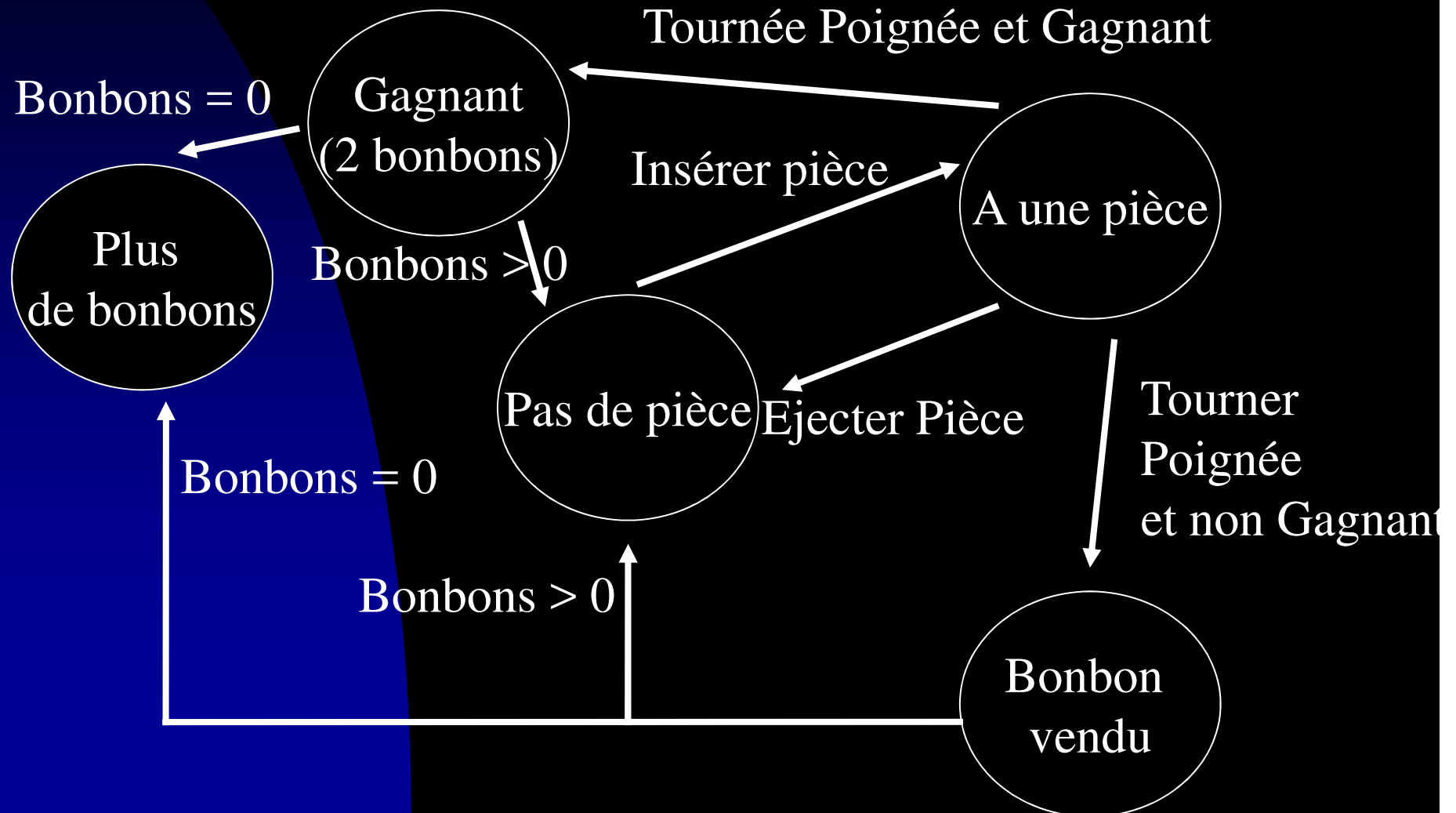
```
public void tournerPoignée() {  
    if (etat == VENDU) {  
        System.out.println(" Pas la peine de tourner deux fois ");  
    }else if (etat == SANS_PIECE) {  
        System.out.println(" Vous n'avez pas payé ");  
    }else if (etat == EPUISE) {  
        System.out.println(" Plus de bonbon ");  
    }else if (etat == A_PIECE) {  
        System.out.println(" Voici un bonbon ");  
        etat = VENDU; nombre - -;  
    }  
}
```

Pattern Etat : un exemple

- Cela fonctionne bien... mais la modélisation n'est pas robuste s'il y a un changement de spécification.
- Supposons la modification suivante :
 - ◆ Tous les 10% du temps (1 fois sur 10), le client reçoit deux bonbons lorsqu'il tourne la poignée.

Pattern Etat : un exemple

- Un état de plus : l'état gagnant



Pattern Etat : un exemple

- Il faut rajouter ce nouvel état et modifier toutes les actions
 - ◆ Ne rien oublier !!
- Une autre modélisation aurait rendu cette modification plus facile
- Cette modélisation est celle du pattern Etat

Pattern Etat

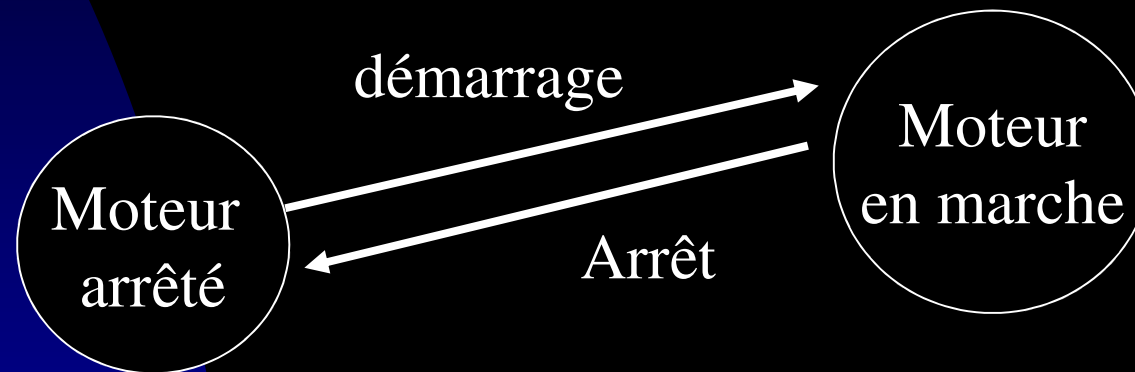
- Un contexte peut être dans n états différents
- Une interface état avec les mêmes méthodes que celles reçues par le contexte
 - ◆ chacun des n états définit une classe implémentant cette interface
 - ◆ un état est muni d'une variable d'instance qui est le contexte

Pattern Etat

- Le contexte contient :
 - ◆ n variables d'instance qui sont les n états possibles
 - ◆ n méthodes qui délivrent respectivement ces n états
 - ◆ une variable d'instance qui est l'état présent
 - ◆ une méthode pour modifier son état présent
- Chaque action adressée au contexte est transmise à son état courant. Etatcourant décide et se modifie tout seul !!

Pattern Etat

- Soit un automate très simple :



Pattern Etat

- Une interface état avec les mêmes méthodes que celles reçues par le contexte

```
interface Etat{  
    public void demarrage();  
    public void arret();  
}
```

Pattern Etat

- Chaque etat possible implémente l'interface (et est muni d'une variable d'instance qui est le contexte)

```
class EtatMarche implements Etat{
    Moteur moteur;

    public EtatMarche(Moteur m){
        moteur=m;
    }

    public void arret(){
        System.out.println("On arrete");
        moteur.modifierEtat(moteur.getArrete());
    }

    public void demarrage(){
        System.out.println("rien a faire.. on tourne");
    }
}
```

Pattern Etat

```
class EtatArret implements Etat{
    Moteur moteur;

    public EtatArret(Moteur m){
        moteur=m;
    }

    public void demarrage(){
        System.out.println("On démarre");
        moteur.modifierEtat(moteur.getMarche());
    }

    public void arret(){
        System.out.println("rien a faire.. a l'arret");
    }
}
```

Pattern Etat

■ La classe Moteur

```
public class Moteur{  
    EtatArret arret;  
    EtatMarche marche;  
  
    Etat etatCourant;  
  
    public Moteur(){  
        arret = new EtatArret(this);  
        marche = new EtatMarche(this);  
        etatCourant = arret;  
    }  
  
    public Etat getArrete(){  
        return arret;  
    }  
  
    public Etat getMarche(){  
        return marche;  
    }  
    ....
```

Pattern Etat

■ La classe Moteur (suite...)

```
....  
public void modifierEtat(Etat e){  
    etatCourant = e;  
}  
  
public void demarrage(){  
    etatCourant.demarrage();  
}  
  
public void arret(){  
    etatCourant.arret();  
}  
}
```


Pattern Etat

- La simulation :

```
public static void main(String[] args) {  
    Moteur m = new Moteur();  
    m.arret();  
    m.demarrage();  
    m.demarrage();  
    m.arret();  
    m.arret();  
    System.out.println("fini");  
}
```