

# PRICMAN

PRICMAN

Variante multi-joueur du célèbre PacMan

ROMAIN BADAMO-BARTHELEMY

CHRISTELLE BODARD

DAVID BUI

ALAN DAMOTTE

ROBIN EUDES

OMBELINE ROSSI

JUIN 2014

# Variante multi-joueur du célèbre PacMan

- ▶ Pac Man collaboratif
- ▶ 4 pacmans : 1 ou 2 pacmans **contrôlables** et respectivement 3 ou 2 pacmans **automatiques** VS fantômes **automatiques**
- ▶ **Plusieurs comportements** disponibles pour les fantômes et les pacmans (3 modes disponibles)
- ▶ Possibilité de gérer les pacmans automatiques par le clic souris

# Variante multi-joueur du célèbre PacMan

- ▶ Fin de partie lorsque toutes les pacgums ont été ramassées
- ▶ L'objectif est d'obtenir le score le plus élevé possible
- ▶ Possibilité de ramasser des **bonus** afin de débloquent des actions spéciales et obtenir des points supplémentaire



Drapeau



Invisibilité

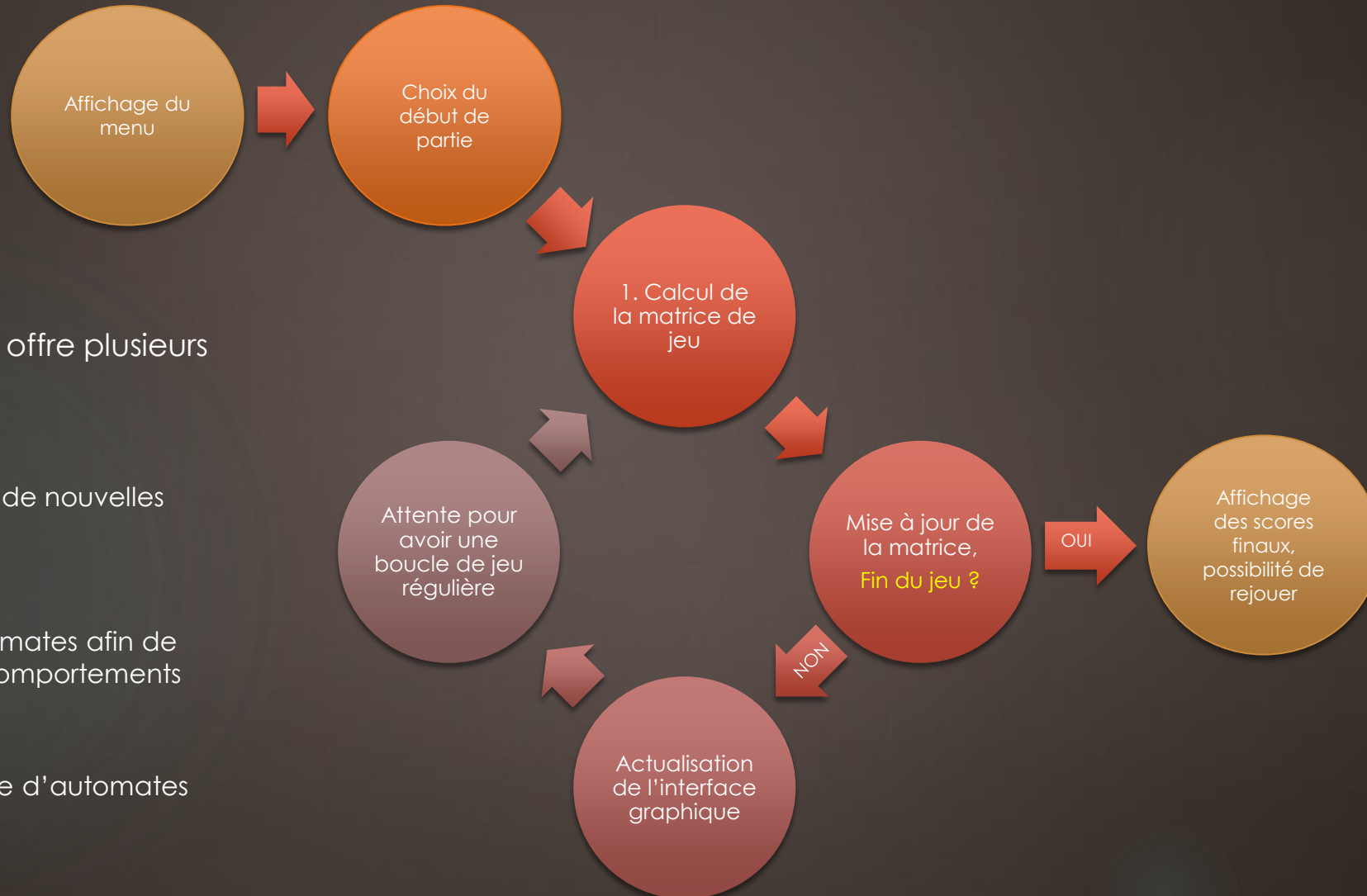


Invincibilité



Ramasse-Gums

# Architecture logicielle



► Cette architecture offre plusieurs possibilités :

- Modifier/ajouter de nouvelles cartes
- Modifier les automates afin de créer d'autres comportements
- Créer davantage d'automates simulés

# Fonctionnement de l'ordonnanceur

L'ordonnanceur se base sur la gestion d'un **seul Thread** de la manière suivante :

**get**(cl : liste des automates associés aux personnages via le fichier xml parsé)

Tant qu'il reste des pacgum sur la carte :

- ▶ Sur un cycle :
  - ▶ **maj\_game**(cl): met à jour les personnages à l'aide de leur comportement
  - ▶ **maj\_matrice** : ramasse les bonus et gère leurs effets. Gère aussi la collision entre pacmen et fantômes.
- ▶ Sur 19 cycles :
  - ▶ Fait avancer chaque personnage d'un pixel si nécessaire (permet un **affichage fluide**)

Pour chaque cycle :

- ▶ **sleep**(durée par cycle prédéterminée)

# Fichier d'entrée

➤ Extrait du fichier ml en entrée : Pacman auto difficile

```
let transition71pa = ((0,"Vers PacGum"),(2,"Fuitniv2"),["Voit Fantome + Invulnerable"],["Fuitniv2"]);;
```

```
let transition72pa = ((1,"Suit"),(2,"Fuitniv2"),["Voit Fantome + Invulnerable"],["Fuitniv2"]);;
```

```
let transition73pa = ((2,"Fuitniv2"),(2,"Fuitniv2"),["Voit Fantome + Invulnerable"],["Fuitniv2"]);;
```

```
let transition74pa = ((2,"Fuitniv2"),(1,"Suit"),["Voit Fantome + Vulnerable"],["Suit"]);;
```

```
let transition61pa = ((0,"Vers PacGum"),(3,"Rejoint"),["A un objectif"],["Rejoint"]);;
```

```
let transition62pa = ((3,"Rejoint"),(3,"Rejoint"),["A un objectif"],["Rejoint"]);;
```

```
let transition63pa = ((3,"Rejoint"),(2,"Fuitniv2"),["Voit Fantome + Invulnerable"],["Fuitniv2"]);;
```

```
let transition64pa = ((2,"Fuitniv2"),(3,"Rejoint"),["A un objectif"],["Rejoint"]);;
```

```
let transitionlist_pad = [transitionverspacgum ; transitionsupm ; transitionfupmd ; transition41pa ; transition61pa ; transition62pa ; transition63pa ;  
transition64pa ; transition71pa ; transition72pa ; transition73pa ; transition74pa];;
```

```
let automatepacautod = ("pad",(0,"Vers PacGum"),[(0,"Vers PacGum")],transitionlist_pad);;
```

# Automates et actions disponibles

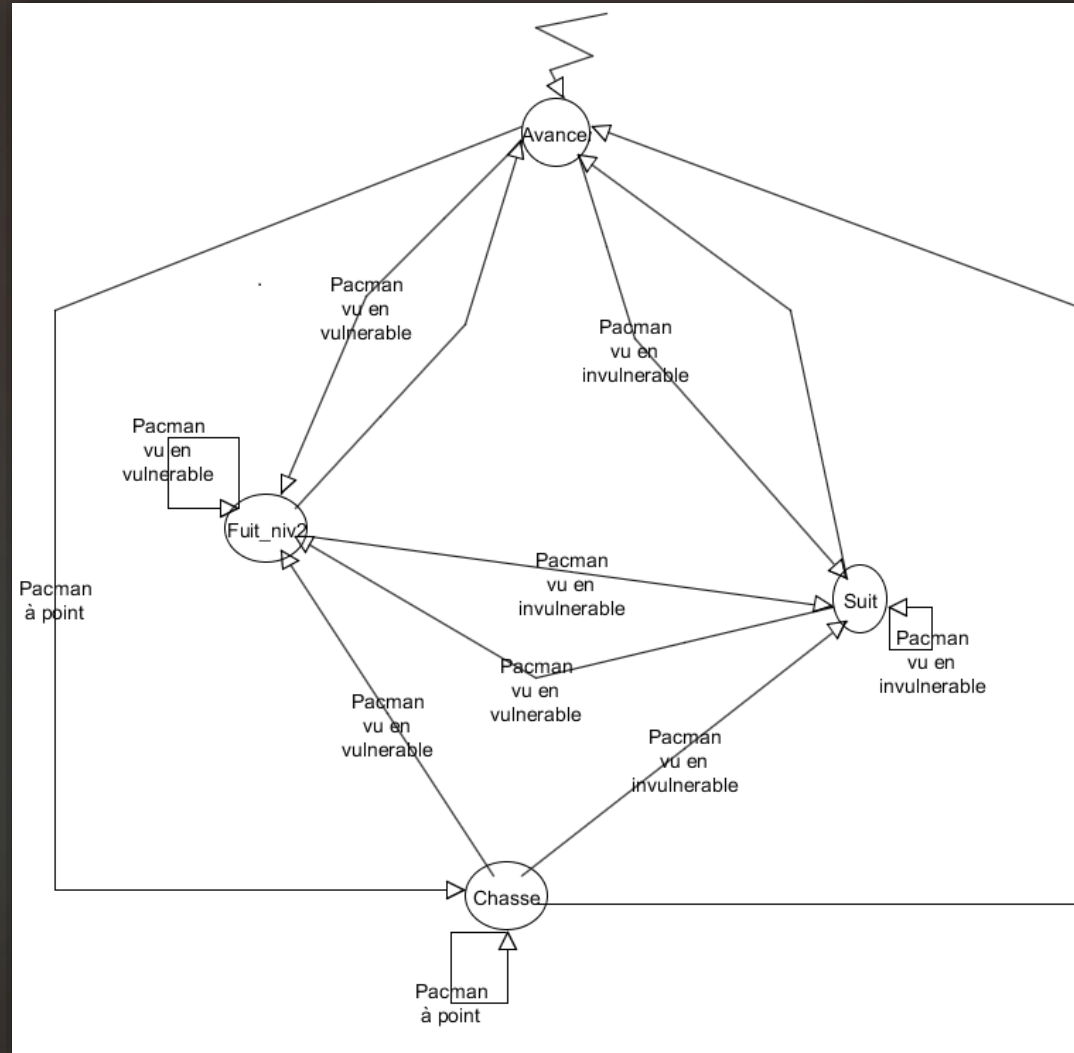
## Actions :

### ► Pacman auto :

- Vers Pacgum
- Fuit
- Fuit\_niv2
- Suit
- Rejoint(l'objectif cliqué)

### ► Fantôme:

- Avance
- Suit
- Fuit
- Fuitniv2
- Chasse



Automate fantôme difficile

## Conditions

### ► Pacman auto :

- Rien
- Voit Fantome + Vulnérable
- Voit Fantome Invulnérable
- A un objectif

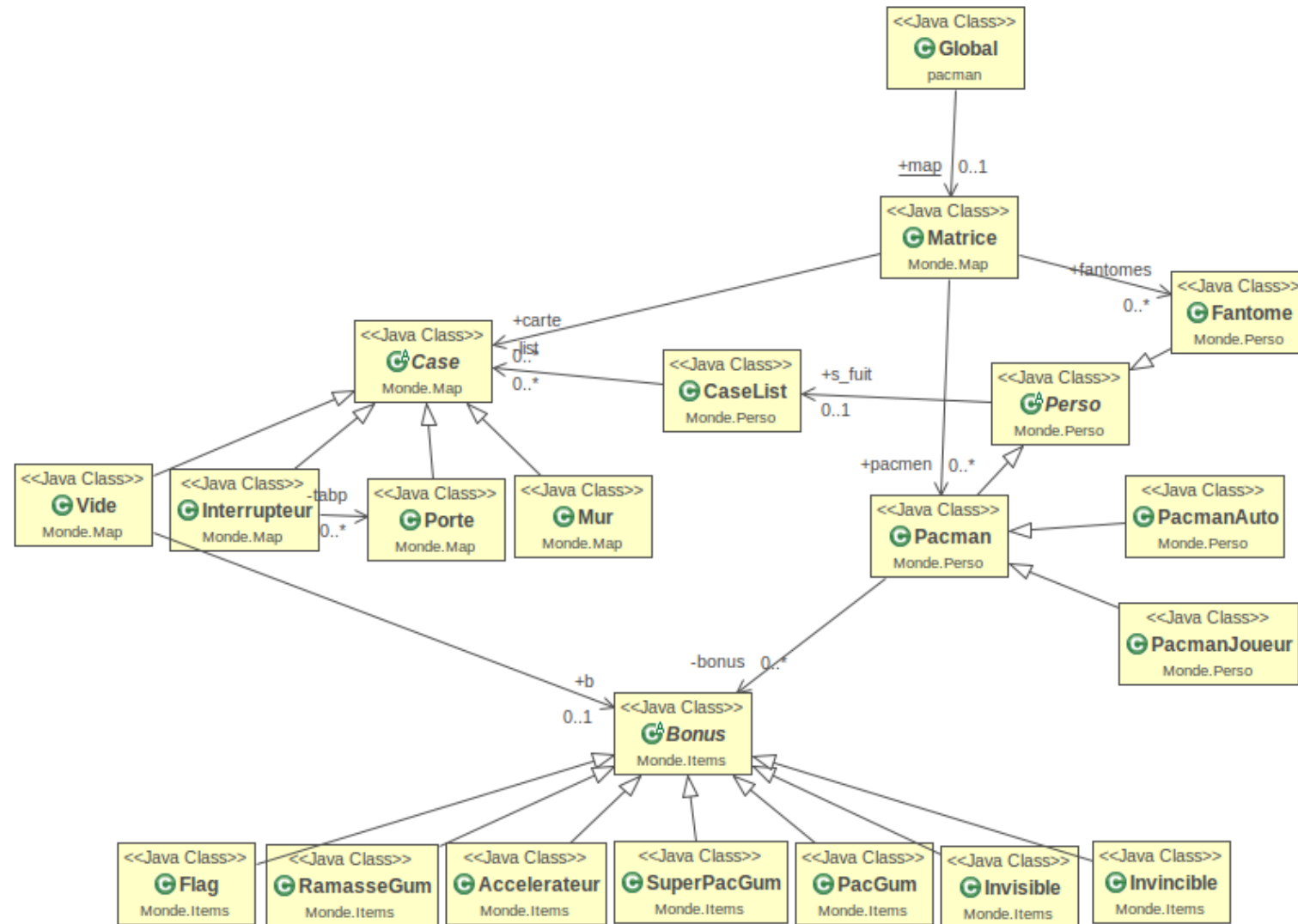
### ► Fantôme:

- Rien
- Voit Pacman + Vulnérable
- Voit Pacman + Invulnérable
- PM à points



# Diagramme des classes (1/2)

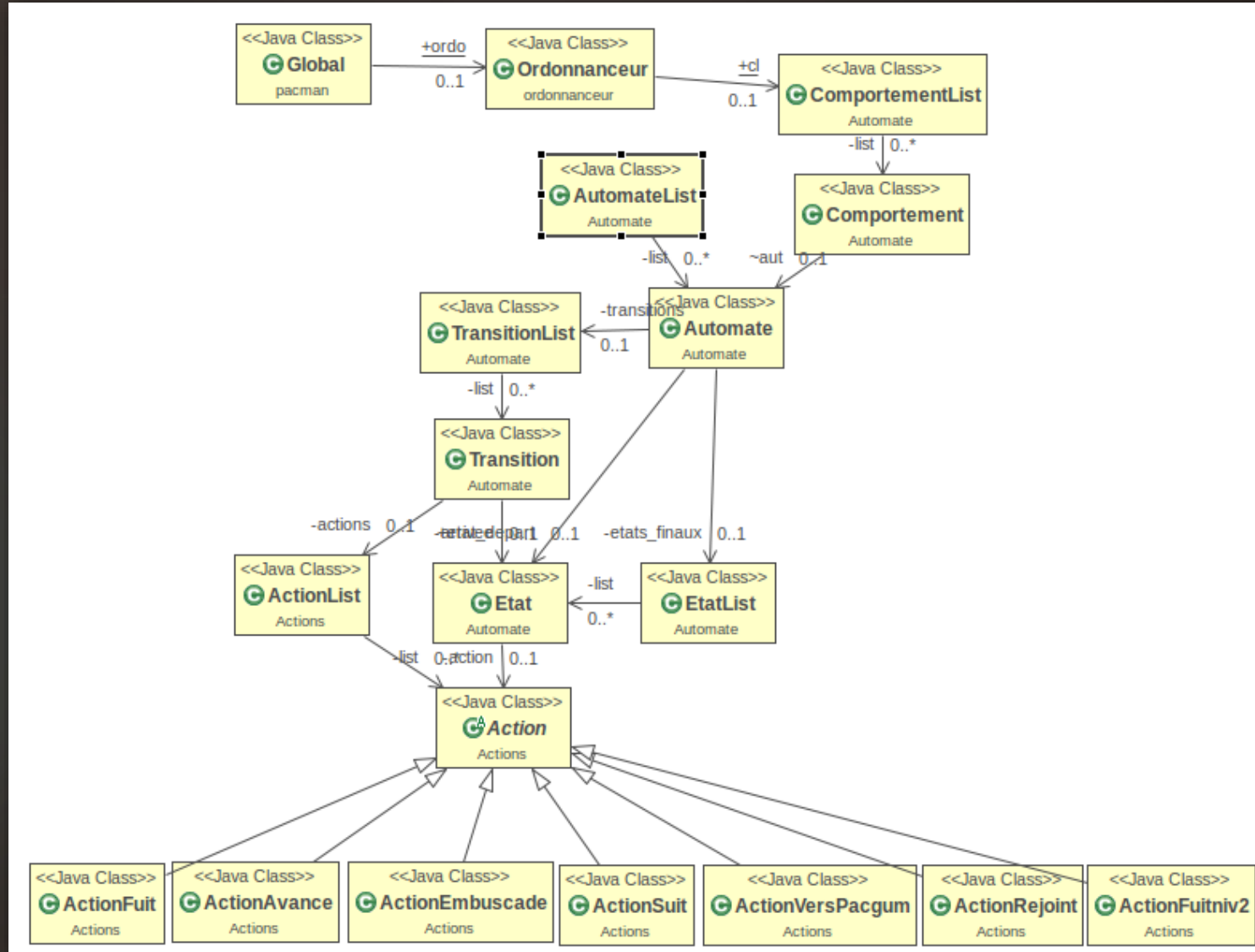
Monde





# Diagramme des classes (2/2)

Partie  
Automate



# Algorithme de l'échappatoire

➤ echappatoire\_haut(x,y)

```
nb_case <= -1
persotrouve <= 0
i <= 0
tant que la case en haut de [x;y] est valide et que persotrouve =0 et que p =-1
    si ce perso est instance de Pacman
        si il y a un fantome sur la case en haut de [x;y]
            persotrouve <= 1;
        fin si
    sinon
        si il y a un pacman sur la case en haut de [x;y]
            persotrouve <= 1;
        fin si
    fin si
    si(persotrouve != 1)
        si (la case à gauche de [x;y] est valide) ou (la case à droite de [x;y] est valide)
            nb_case <= i
        fin si
    fin si
    i++;
fin tant que
retourner nb_case;
```

# Principales difficultés rencontrées

- ▶ Gestion de l'affichage des déplacements (fluidité)
- ▶ Automate des fantômes et des pacmans intelligents
- ▶ Embuscade : Algorithme de Dijkstra difficilement implémentable en dynamique
  - ▶ Une alternative qui consiste à se diriger vers la zone où se trouve actuellement le pacman

# Répartition du travail

## ▶ Alan

- ▶ Interface graphique
- ▶ Gestion événementielle boutons de jeu et menu
- ▶ Fenêtre de dialogue
- ▶ Gestion des scores

## ▶ David et Romain

- ▶ Gestion de l'architecture logicielle
- ▶ Gestion des bonus et insertions
- ▶ Création des sprites PM, FT, bonus
- ▶ Tests

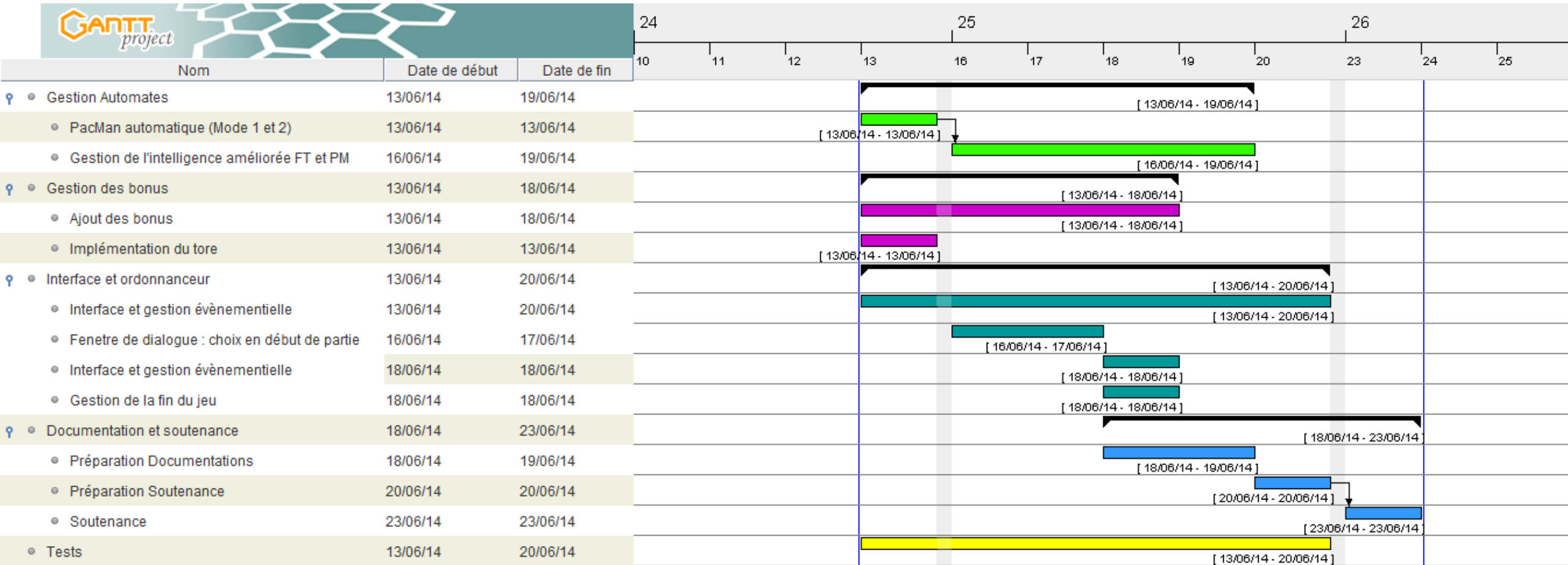
## ▶ Christelle et Ombeline

- ▶ Parser Ocaml vers XML
- ▶ Parser XML vers Java
- ▶ Gestion des différents niveaux d'automates
- ▶ Actions et conditions

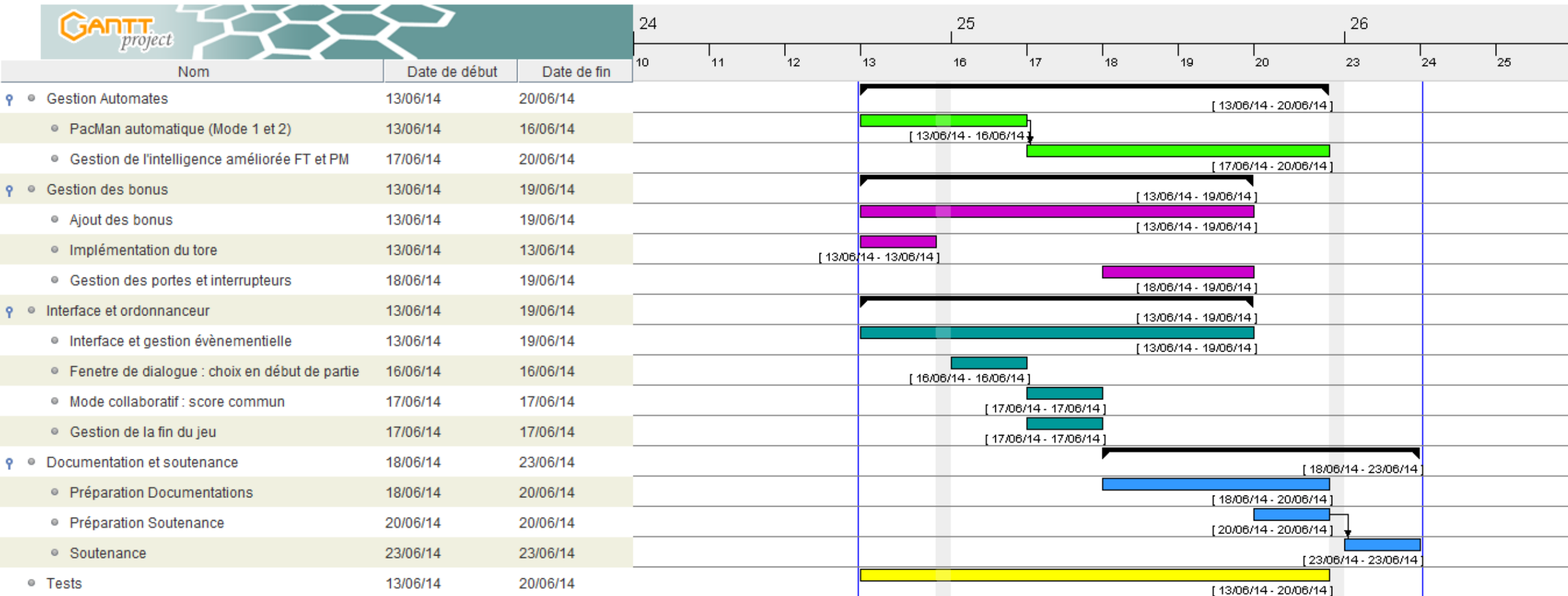
## ▶ Robin

- ▶ Ordonnanceur
- ▶ Débogage partie Java
- ▶ Gestion événementielle

# Planning initial (après prototypage)



# Planning réel (après prototypage)



# Respect du contrat

## ► Contrat

- (1 ou 2) PM contrôlables + (1 ou 2 PM automatisés) vs FT OK
- Fin de partie lorsque les billes ont été ramassées OK
- Actions : avancer, chasse, suivre, tue, stop, mourir (mises à jour) OK
- Conditions : contact, voit, perd, changement direction (mises à jour) OK
- Pac Man secondaire automatique :
  - aléatoire, OK
  - éviter les fantômes, OK
  - dirigé par un objectif (indiqué par un clic souris) OK

## ► Extensions

- mode embuscade organisée par les FT avec calcul du meilleur chemin pour bloquer PM en prenant un chemin différent des autres fantômes
- ➡ Les fantômes se dirigent dans la direction du pacman qui a été vu et qui a le plus de points OK