

# Génie Logiciel Avancé

## Projet

Stefano Zacchioli  
zack@pps.jussieu.fr

Laboratoire PPS, Université Paris Diderot - Paris 7

17 mars 2011

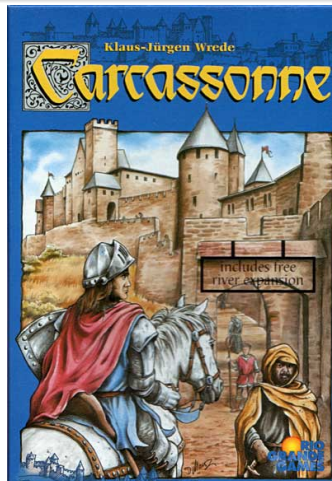
URL <http://upsilon.cc/zack/teaching/1011/g1a/>  
Copyright © 2011 Stefano Zacchioli  
License Creative Commons Attribution-ShareAlike 3.0 Unported License  
<http://creativecommons.org/licenses/by-sa/3.0/>



# Sujet : Carcassonne

Réaliser une version électronique du jeu de société **Carcassonne**

- construction d'un paysage médiéval par la pose de **tuiles**
- jeu de **stratégie** au **tours** par tours, pour 2-6 joueurs
- le paysage est partagé entre joueurs
- chaque tuile impose de **contraintes** sur la pose de tuiles suivantes
- beaucoup d'**extension** qui ajoutent *nouvelle règles et tuiles*



1 Règles

2 Le projet

# Principe

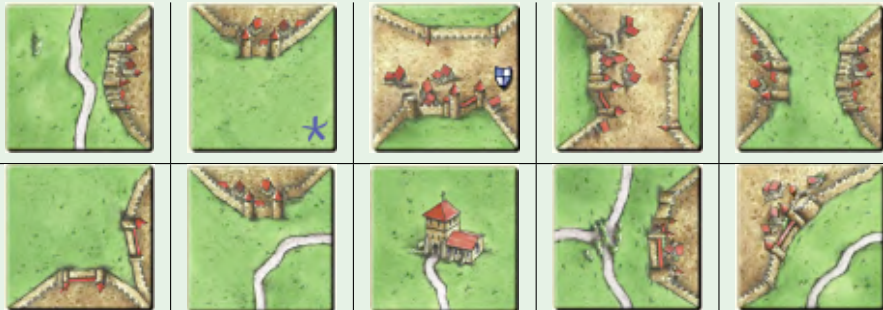
- jeu de pose de **tuiles** pour 2–6 joueurs
- le jeu commence avec une seule tuile visible, les autres tuiles sont cachées dans la pioche
- chacun à son tour, les joueurs piochent une tuile et tentent de la placer respectant les **contraintes** imposées par le paysage
- après le placement de la tuile—et avant le tour du joueur successif—le joueur *peut* placer un **pion** sur une partie de la tuile posée

# Tuiles

Les tuiles sont carrées et peuvent imposer 3 types des contraintes sur leur (futurs) voisines :

- 1 être une **ville**
- 2 être un **champ**
- 3 être un **chemin**

## Exemple (Tuiles)



# Contraintes

- Une tuiles doit être posée adjacente à d'autres tuiles déjà posées.
- Une tuile doit respecter les contraintes des toutes tuiles adjacentes.

Placement correct



Placement incorrect



# Pions

Les pions peuvent être posés sur une des partie de la tuile posée et plus particulièrement sur :

- un(e partie de) **chemin**
- une (partie de) **ville**
- un(e partie de) **champ**
- un **cloître**

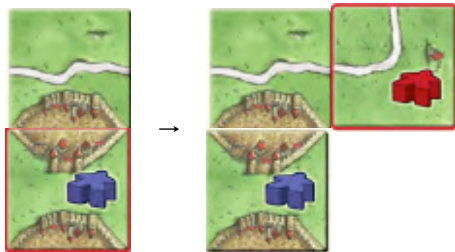
Un pion ne peut pas être posé si l'élément de destination choisi est partie (en suivant ses adjacences) d'un chemin/ville/champ ou un autre pion est déjà présent.

# Mais...

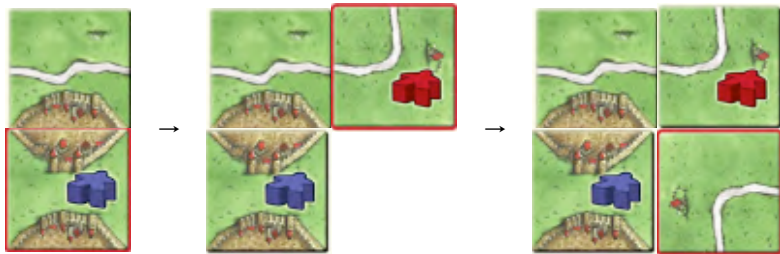




# Mais...



# Mais...



# Points

## Pendant le jeu

Quand un des éléments suivants est complétée, les pions sont retournées aux joueurs correspondants et le **propriétaire(s)** de l'élément compte des points :

- chemin** 1 point pour chaque tuile qui compose le chemin
- ville** 2 point pour chaque tuile qui compose la ville + 2 point pour chaque tuile "bouclier" qui est partie de la ville
- cloître** 9 points (un cloître est complet quand il est contourné)

### À la fin du jeu

**chemin** 1 point pour chaque tuile

**ville** 1 point pour chaque tuile + 1 point pour chaque bouclier

**cloître** 1 point pour le cloître + 1 point pour chaque tuile dans le périmètre du cloître

**champ** 3 point pour chaque ville complète qui est adjacent au champ (les pions placés sur des champs restent jusqu'à la fin du jeu)

### Qui est le propriétaires d'un élément?

- si l'élément est occupé par 1 seul pion → le propriétaire du pion
- si l'élément est occupé par plusieurs pions → le jouer qui a la majorité des pions
  - ▶ voir slide "*mais...*"
- en cas d'*ex equo* → tous les joueurs qui ont la majorité

# Extensions

Le jeu a connu un très gros succès, induisant une série d'extensions. Chaque extension ajoute nouvelles (types de) tuiles, nouvelles règles, et d'autre *gadget* (e.g. éléments de paysage physiques, nouveau pions, etc.).

## Exemple (La fleuve)



- le jeu commence par le spring, les tuiles fleuve sont posées avant les autres

## Extensions (cont.)

Le jeu a connu un très gros succès, induisant une série d'extensions. Chaque extension ajoute nouvelles (types de) tuiles, nouvelles règles, et d'autre *gadget* (e.g. éléments de paysage physiques, nouveau pions, etc.).

### Exemple (Marchands et bâtisseurs)

- nouveau pion *builder*, peut être posé, au lieu de poser un pion, sur un élément dont le joueur a déjà posé un pion
- nouveau pion cochon similaire pour les champs, qui fait valoir à la fin du jeu les villes 4 points au lieu de 3...
- toutes ajoutées des tuiles futur au même élément, permettes au joueur d'avoir un tour extra
- des nouveau types des boucliers sont ajoutés pour les villes, des trois types, le jouer qui complète un ville (pas le propriétaire) collection les boucliers. À fin du jeu, qui a la majorité des boucliers dans chaque type des bouclier, compte 10 points extra.

Les info fournis ici sur les règles du jeu sont seulement un introduction et pas une référence complète !

- [http://en.wikipedia.org/wiki/Carcassonne\\_\(board\\_game\)](http://en.wikipedia.org/wiki/Carcassonne_(board_game))
- <http://www.ludism.fr/regles/?Titre=Carcassonne>
- <http://upsilon.cc/~zack/teaching/1011/g1a/carcassonne/>
  - ▶ si vous trouvez d'autre matériel extra, librement re-distribuable, je serais ravis de l'ajouter
- le jeu lui même (au cas ou...)



1 Règles

2 Le projet

# Taches

- 1 spécification des besoins (optionnel)
- 2 conception (à objets)
- 3 implémentation (d'un sous-ensemble) des sous-systèmes :
  - ▶ moteur de jeu
  - ▶ gestionnaire d'extensions
  - ▶ extensions
  - ▶ intelligence artificielle
  - ▶ client/serveur pour distribuer la computation sur réseau
  - ▶ interface(s) utilisateur
- 4 documentation (utilisateurs<sup>1</sup> et développeurs)
- 5 présentation (soutenance)

---

## 1. optionnel

# Spécification des besoins et conception

- diagrammes des cas d'utilisation (optionnel)
- diagrammes des paquets
- diagrammes des classes
- diagrammes de déploiement (optionnel)
- diagrammes de séquence ou nécessaire à comprendre l'interaction entre classes

*voir cours 2 et 3*

# Implémentation

## Sous-systèmes

- |                            |                                |
|----------------------------|--------------------------------|
| ① moteur de jeu            | ⑤ intelligence artificielle    |
| ② interface(s) utilisateur | ⑥ client pour jouer en réseau  |
| ③ gestionnaire d'extension | ⑦ serveur pour jouer en réseau |
| ④ extension(s)             |                                |

Chaque group devra rendre :

- une implémentation originale de (1) et (3)
- une implémentation originale d'une **module extra** à choix entre : (2), (4), (5), (6), (7)
  - ▶ l'interface utilisateur (2) est à choix : GUI, interface textuelle, ...
  - ▶ l'extension (4) est à choix entre les extensions existants
- deux modules extra implémentés par d'autres groupes, mais **intégrés** avec le projet du group
  - ▶ chaque projet devra contenir au moins une interface utilisateur

# Coordination

Un projet qui contient pas des modules extra implémentés par d'autres groupes est un projet incomplet. Donc vous devrez **collaborer** beaucoup pour partager modules entre groups différents (p.ex. pour définir des interfaces communes).

Pour aider la coordination entre groups différentes une nouvelle **liste de diffusion** à été crée :

<https://listes.sc.univ-paris-diderot.fr/sympa/info/m1g1-projet>

L'inscription est fortement conseillé pour tous.  
Est à vous d'animer les discussions.

**utilisateur** comment utiliser l'interface, comment obtenir et charger des extensions, comment jouer en réseau, etc.

**développeur** comment écrire des extensions, comment maintenir et adapter le code en futur, protocole de communication réseau, etc.

# Contraintes des processus

- spécification et conception à objet avec UML 2.0 comme notation
- développement dans un langage de programmation à choix
  - ▶ mais à objet
- tests unitaires de chaque classe
- tests d'intégration de système
- développement à l'aide d'un gestionnaire de version
  - ▶ dont l'historique ferai partie de l'envoi du projet

- projet à développer par group de 2 personnes



# Dates

10 mars 2011 présentation du projet

10 mars 2011 1er TD de projet (questions/réponses)

17 mars 2011 déclaration des groups, date limite

17 mars 2011 2e TD de projet (questions/réponses)

... 2 mois ...

15 mai 2011 date limite pour rendre les projets

26 mai 2011 examen

début juin 2011 soutenances

*juste après l'examen; à définir*