

# Développement d'un jeu Bomberman sous Android et iOS

## TER

KLOB

K.COUSEIN L.PITIOT O.BONVILA B.TARDIEU  
Encadrant: L.DERUELLE

Université Montpellier II

27 mai 2011

# Sommaire

- 1 Introduction
- 2 Présentation
- 3 Application
- 4 Réutilisabilité
- 5 Discussion
- 6 Conclusion

# Introduction

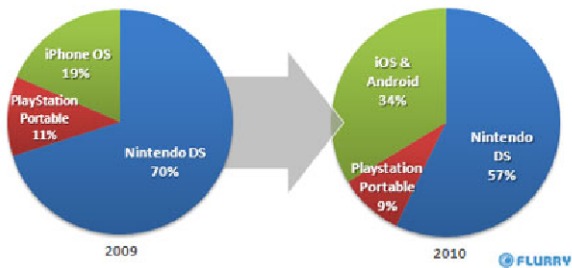


# Présentation

## Les jeux sur smartphone

Le marché des jeux vidéos sur console portable connaît une réelle expansion :

U.S. Portable Game Software by Revenue



# Présentation

## Android

Le système d'exploitation possède :

- Noyaux linux pour exploiter le matériel
- Bibliothèques connues et open source (OpenGL ES, SQLite,...)
- Machine virtuelle Java (Dalvik virtual machine)
- API Java riche (package de Java SE, open source ou spécifique au système)



# Présentation

## Android

Les possibilités de développement sont :

- Langage principal Java, développement en C/C++ possible
- Kit de développement multiplateforme
- Développement sur téléphone ou sur émulateur
- Déploiement des applications peu coûteux



# Présentation iOS

Le système d'exploitation possède :

- Noyau dérivé de Mac OS X
- Bibliothèques connues et open source (OpenGL ES, SQLite,...)
- Pas de machine virtuelle. Code compilé en C.
- API Objective-C riche (Core OS, Cocoa Touch,...)



# Présentation iOS

Les possibilités de développement sont :

- Langage principal Objective-C, développement en C possible
- Kit de développement disponible sur Mac OS seulement
- Développement sur emulateur seulement.
- Déploiement des applications coûteux



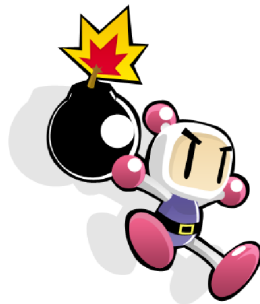


# Présentation

## Bomberman

### Histoire

- Jeu d'action.
- Première apparition en 1987.
- Développé par Hudson Soft.
- Développé sur plusieurs consoles.
- Succès grâce au mode multijoueur sur certaines consoles.



# Présentation

## Bomberman

### Principe :

- Le joueur incarne un poseur de bombes.
- But du jeu : détruire ses ennemis.
- Multiples bonus (Bonus de vie, de bombes, de vitesse,...).
- Multiples malus (Obligation de poser des bombes,...).



# Présentation

## Rapport avec l'enseignement

Ce TER nous a permis de mettre en application les connaissances acquises dans nos parcours d'enseignements.

### Aspect technique

Intelligence artificielle  
Communication mobile-serveur  
Serveur d'application  
Conception de logiciel

### Aspect pédagogique

I2A  
CASAR  
DIWEB  
Génie Logiciel

# Application

## Chargement de l'application

Premier chargement :

- 1 Création de la base de données
- 2 Récupération de toutes les ressources dans le fichier XML
- 3 Instanciation des objets grâce aux ressources récupérées
- 4 Copie des cartes dans le répertoire d'installation du téléphone
- 5 Initialisation de la base de données
- 6 Affichage de la page de création de compte local

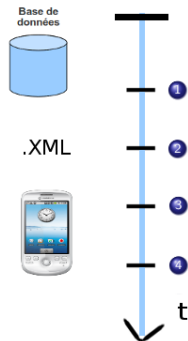


# Application

## Chargement de l'application

Chargement standard :

- 1 Chargement de la base de données
- 2 Récupération de toutes les ressources dans le fichier XML
- 3 Instanciation des objets grâce aux ressources récupérées
- 4 Instanciation du dernier utilisateur



# Application

## Les menus

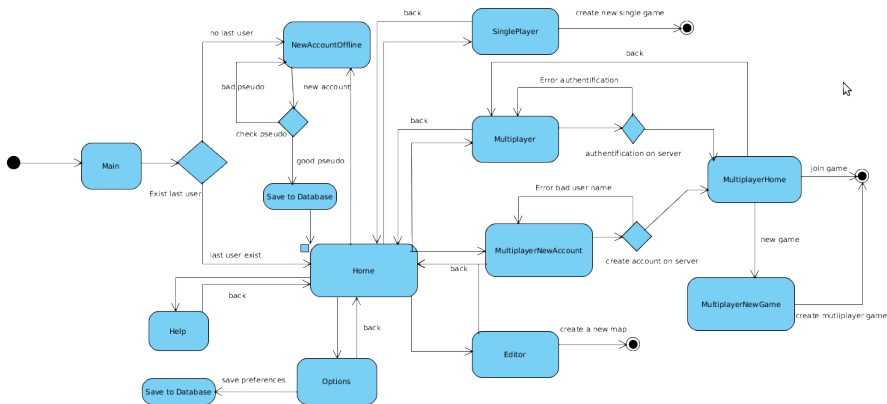
### Objectifs de conception :

- Interface claire
- Facile d'utilisation
- Navigation intuitive
- Ergonomique



# Application

## Modélisation



# Application

## Types d'objets

Deux types d'objets :

- Non animés (tile) : une seule et unique image



- Animés (sprite) : une sequence d'images (animation)





# Application

## Tile mapping

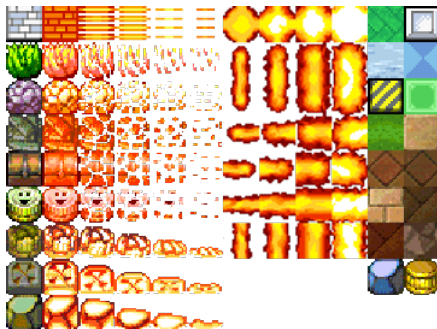
Pourquoi ?

- 1 Tire ses origines des jeux des années 80
- 2 Faible consommation des ressources
- 3 Performances des smartphones limitées

# Application

## Tile Mapping

Principe :



# Application

## Gestion des ressources

### XML

- Code portable
- Syntaxe extensible (Générique)
- Ajout/suppression d'éléments facile

```
<objects>  
  <undestructible name="grass1" hit="0" level="0" fireWall="0" damages="0">  
    <animation name="idle" canLoop="false" sound="">  
      <framerect top="270" left="120" bottom="300" right="150" delayNextFrame="0" />  
    </animation>  
  </undestructible>  
</objects>
```

# Application

## Editeur de cartes

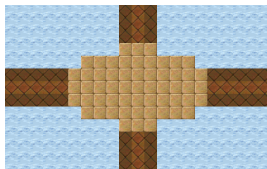


# Application

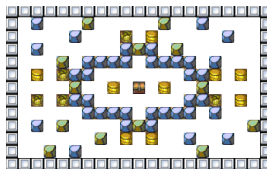
## Moteur de rendu (Editeur de cartes)

Deux matrices :

1<sup>er</sup> niveau



2<sup>ème</sup> niveau



# Application

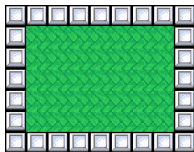
## Jeu



# Application

## Moteur de rendu (Jeu)

Bitmap d'objets  
inanimés

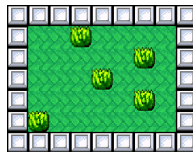


+

Table de hashage  
d'objets animés

=

Resultat



# Application

## Moteurs de rendu

Complexité en nombre d'objets à afficher

	Editeur de carte	Jeu
Meilleur des cas	337	1
Pire des cas	534	229

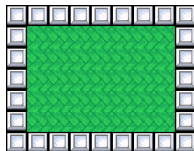


FIGURE: Meilleur des cas



# Application

## Moteur Physique

Carte des colisions :

# Application

## Intelligence artificielle

### Niveaux de difficulté

- 1 Facile
- 2 Moyen
- 3 Difficile



# Application

## Pathfinding

### Algorithme de parcours en largeur

- 1 Pas de case d'arrivée nécessaire
- 2 Tous les chemins possibles
- 3 Premier chemin trouvé
- 4 Rapidité

# Application

## Pathfinding

### Algorithme A\*

- ① Heuristique (de Manhattan)
- ② Coût de déplacement
- ③ Premier chemin trouvé
- ④ Rapidité (Dijkstra)

	A	B	C	D	E	F	G
1	94 B2	80 B2	74 C2				
2	24 70	20 60	24 50				
3	74 B3	60 B3	54 B3				
4	14 60	10 50	14 40				
5	60 B3	F Père	40 B3		82 F4	F4	82 F4
6	10 50	G H	10 30		72 10	68 0	72 10
7	74 B3	60 B3	54 B3		74 E5	68 E5	88 F4
8	14 60	10 50	14 40		54 20	58 10	68 20
9	94 B4	80 B4	74 C4	74 C5	74 D5	74 E5	102 F4
10	24 70	20 60	24 50	34 40	44 30	54 20	72 30

# Application

## Outils

- Servlets
- Serveur d'application
- JSON
- MySQL



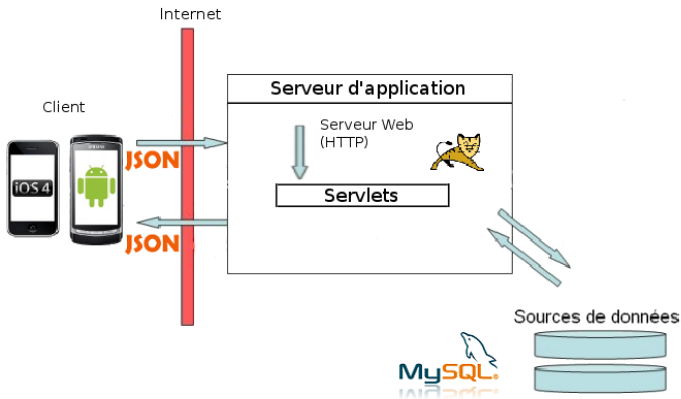
### Remarque :

- Pas de moteurs de rendu
- Pas d'intelligence artificielle
- Moteur physique

# Application

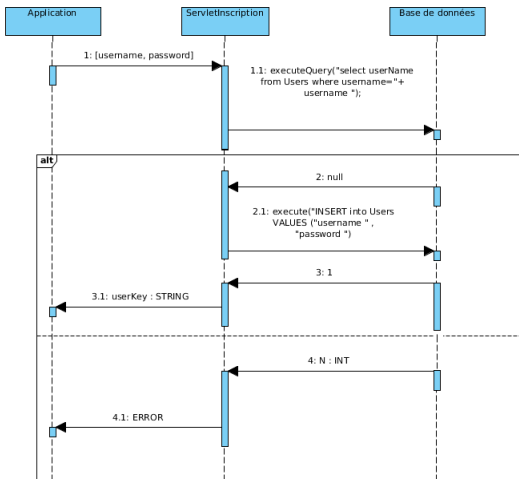
## Principe

Schéma de fonctionnement :

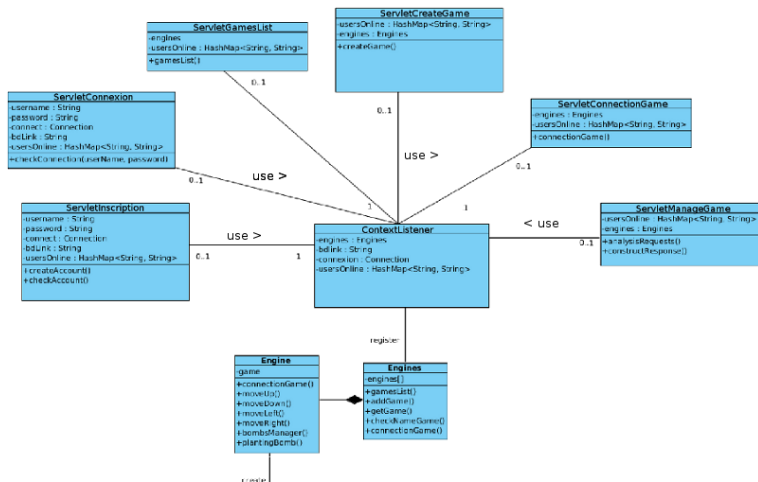


# Application

## Exemple



# Application Modélisation



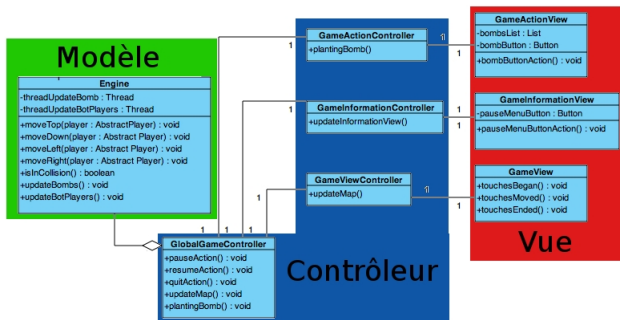


# Réutilisabilité

## Généralités

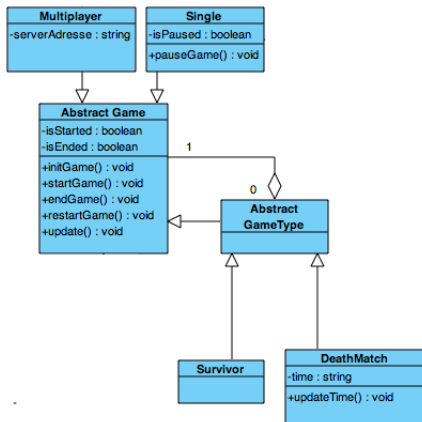
- Documentation et code en anglais
- Internationalisation du jeu

# Réutilisabilité MVC



# Réutilisabilité

## Nouveaux types de parties



- Design pattern décorateur
- Permet l'extension du modèle

# Réutilisabilité

## Personnalisation

XML L'utilisation du XML permet d'ajouter ou de personnaliser facilement :

- Des objets (bonus, blocs, bombes, etc ...)
- Des images (themes)
- Des sons

# Réutilisabilité

## Fonctionnalités

- Nouvelles servlets = nouvelles fonctionnalités
- Même hiérarchie de classes que du côté client

# Discussion

## Difficultés

Android	iOS	Serveur
	Ressources limitées	Déploiement local
	Nouvelle plate-forme	Servlets
	Nouveau langage (Objective-C)	Communication avec la base de données
Multi-touch	Gestion manuelle de la mémoire	

# Discussion

## Problèmes

Android et iOS :

- Tester l'application
- OpenGL ES

Serveur :

- Déploiement internet

# Discussion

## Améliorations

- Mode histoire
- Ajout de bonus / malus
- Rajout de types de parties
- Gestion des scores
- Nouvelles bombes



# Conclusion

Apports en relation avec nos parcours d'enseignement

I2A	CASAR	DIWEB	GL
Recherche opérationnelle	Communication mobile-serveur	Servlet	Conception d'une application
A*	Mise en place d'un serveur	BDD	MVC
Parcours en largeur	Sécurisation du réseau	XML	Design pattern décorateur
Moteur de jeu		IHM ergonomique	

# Conclusion

Ce que cela nous a apporté

- Découverte de la programmation mobile (SDK Android, SDK iOS).
- Apprentissage d'un nouveau langage (Objective-C).
- Découverte de la programmation de jeux vidéos.
- Communication mobile-serveur.
- Travail en groupe.