




14/06/2015

Un moteur de jeu 2D en Java

Plan

2

- **Moteur** 
- Module Core
- Module Game
- Module Network
- Editeur

Moteur

3

lionengine-core

lionengine-
core-awt

lionengine-
core-swt

lionengine-
core-
android

lionengine-
game

lionengine-
network

lionengine-
audio-
(wav, midi,
sc68)

Moteur

4

- API « bas niveau »
 - ▣ Manipulation des ressources
 - Visuelles (images, sprites, animations, parallaxe)
 - Sonores (sons et musiques)
 - Fichiers (binaires et XML)
 - Périphériques (curseur système / « in-game »)
 - ▣ Environnement graphique
 - Résolution écran (+filtrage: bilinéaire, HQ2X, HQ3X)
 - Modes de rendu (fenêtré, plein écran, applet)
 - Gestion du « frame rate »
 - Gestion des séquences (intro, menu, scene...)

Moteur

5

- API « haut niveau »
 - ▣ Abstraction de premier niveau
 - Classes de base orientées jeux-vidéo généraux
 - Rutines de base implémentées et redéfinissables
 - Architecture souple et modulaire
 - Outils standards
 - ▣ Abstraction de deuxième niveau
 - Interfaces et classes de base dédiées à certains types
 - Gestion des collisions
 - Pathfinding
 - ...

Moteur - Modules

6

- Un module est présent sous la forme d'un JAR
 - ▣ Inclusion aisée des modules sur un projet
 - ▣ Nécessité d'inclure la partie centrale d'abord

- Chaque module
 - ▣ Dépend du module principal (`lionengine-core`)
 - ▣ Propose une base abstraite (`architecture de base`)
 - ▣ Est redéfinissable selon les besoins, en tout point
 - ▣ Est compatible avec d'autres modules
 - ▣ Respecte la même structure que la partie centrale

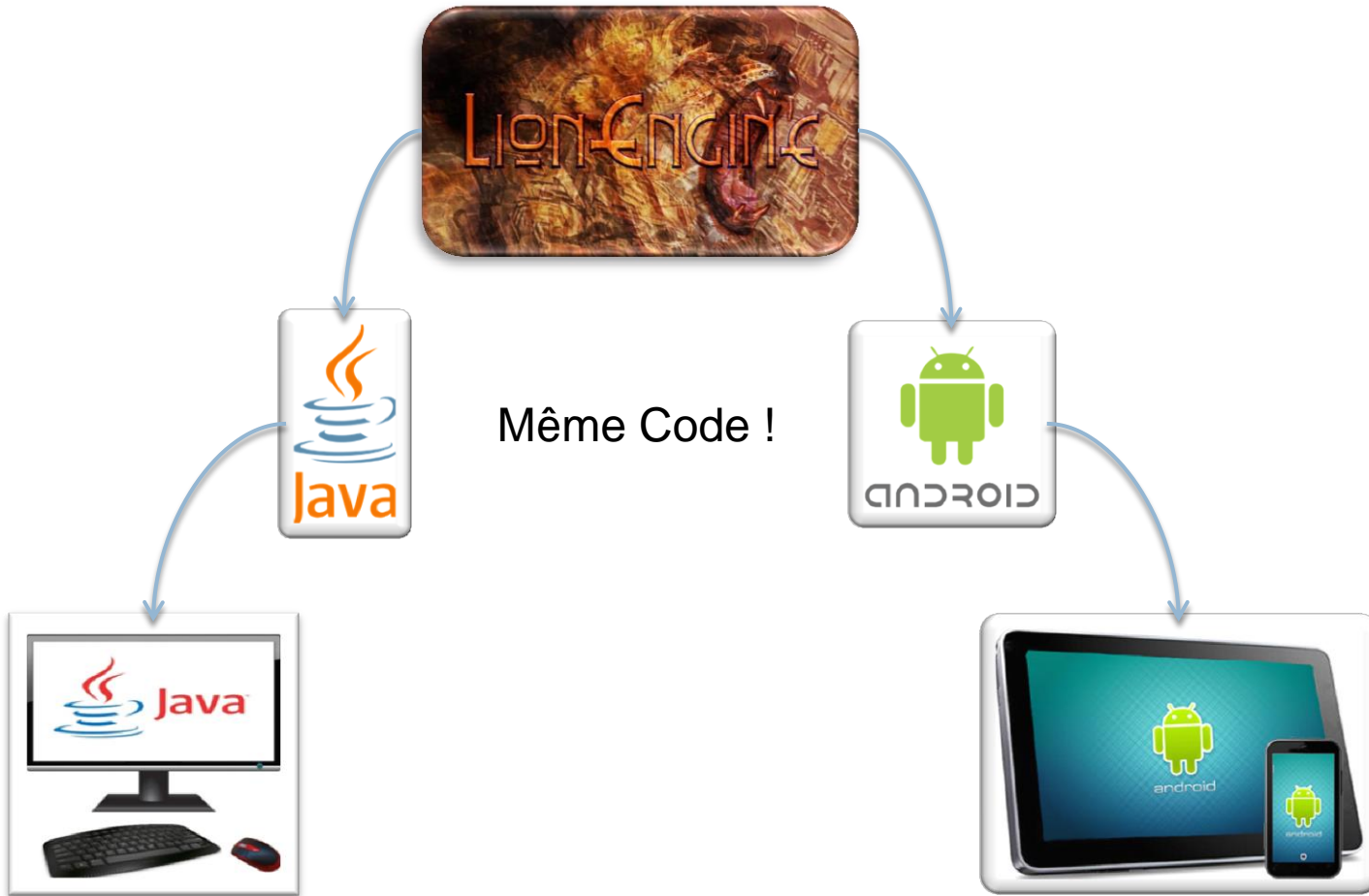
Moteur - AWT, SWT, Android

7

- Le moteur ne dépend d'aucune implémentation bas niveau directe
- 3 modules au choix
 - ▣ lionengine-core-awt
 - Dédié aux jeux sur PC, fonctionnant en fenêtré, plein écran, et applet en utilisant Java2D
 - ▣ lionengine-core-swt
 - Dédié aux jeux sur PC, fonctionnant en fenêtré et plein écran en utilisant SWT
 - ▣ lionengine-core-android
 - Dédié aux jeux pour smartphone et tablette, avec Android 2.3.3 au minimum


Moteur - Utilisation

8



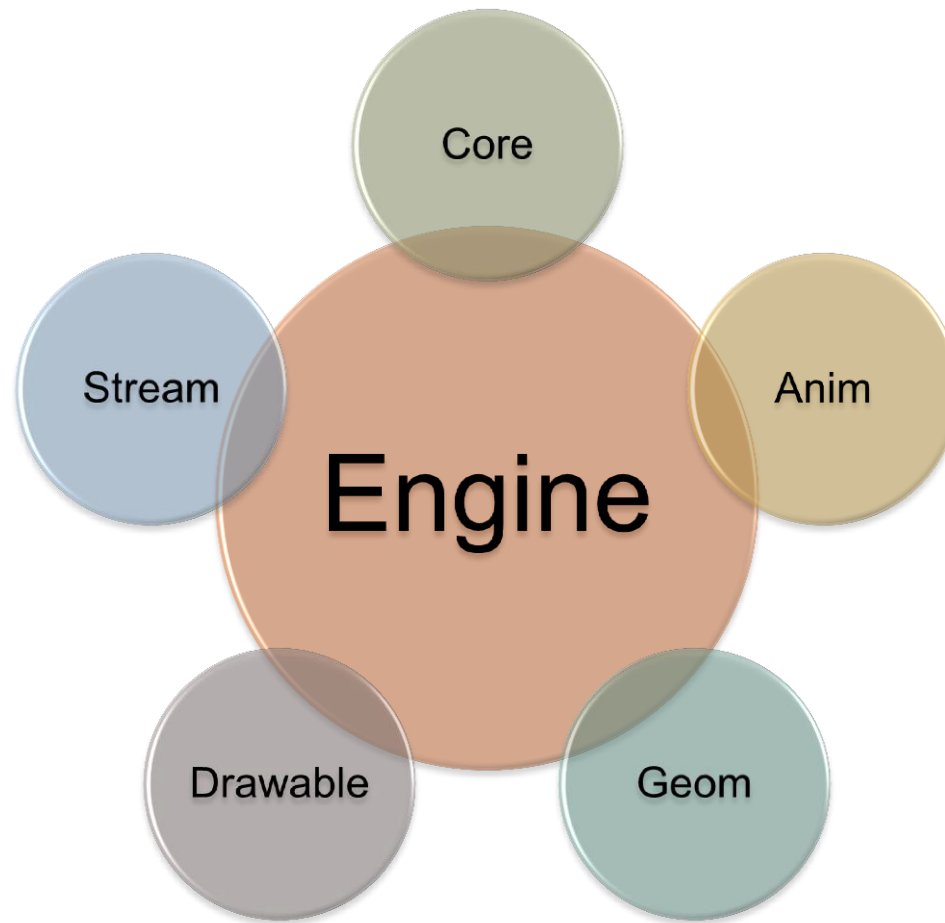
Plan

9

- Moteur
- **Module Core** 
- Module Game
- Module Network
- Editeur

Module Core

10



Module Core

11

- ❑ **Situé à partir du package:** `com.b3dgs.lionengine`
- ❑ **Principaux packages / classes**
 - ❑ **anim** (*Animator, Animation, AnimState*)
 - ❑ **core** (*Engine, Config, Graphic, Sequence...*)
 - ❑ **drawable** (*Image, Sprite, SpriteTiled, SpriteFont...*)
 - ❑ **stream** (*FileReading, FileWriting, XmlFactory, XmlNode*)
 - ❑ **geom** (*Coord, Rectangle, Polygon...*)

Module Core - Sequence

12

- **Squelette de base**
 - ▣ `load();`
 - ▣ `update(double extrp);`
 - ▣ `render(Graphic g);`
 - ▣ `onTerminate(); // Optionnel`
- **Gestion du nombre d'images par seconde**
- **Gestion de l'extrapolation** ('machine independant')
- **Modes d'affichage:** plein écran, fenêtré, applet

Module Core - Sequence

13



load

- Initialisation des variables
- Chargement des ressources

update

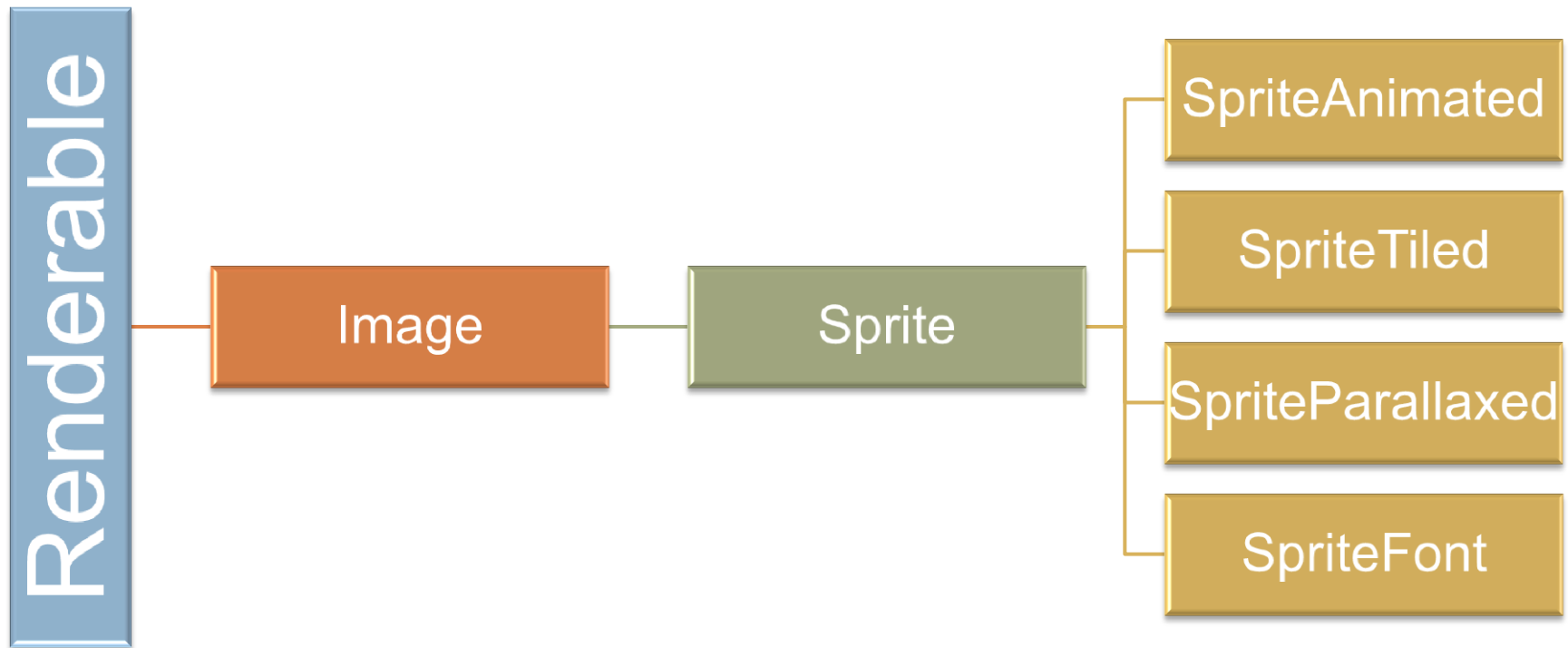
- Mise à jour des variables
- Mise à jour des composants

render

- Rendu dans un buffer
- Affichage du buffer à l'écran

Module Core - Drawable

14



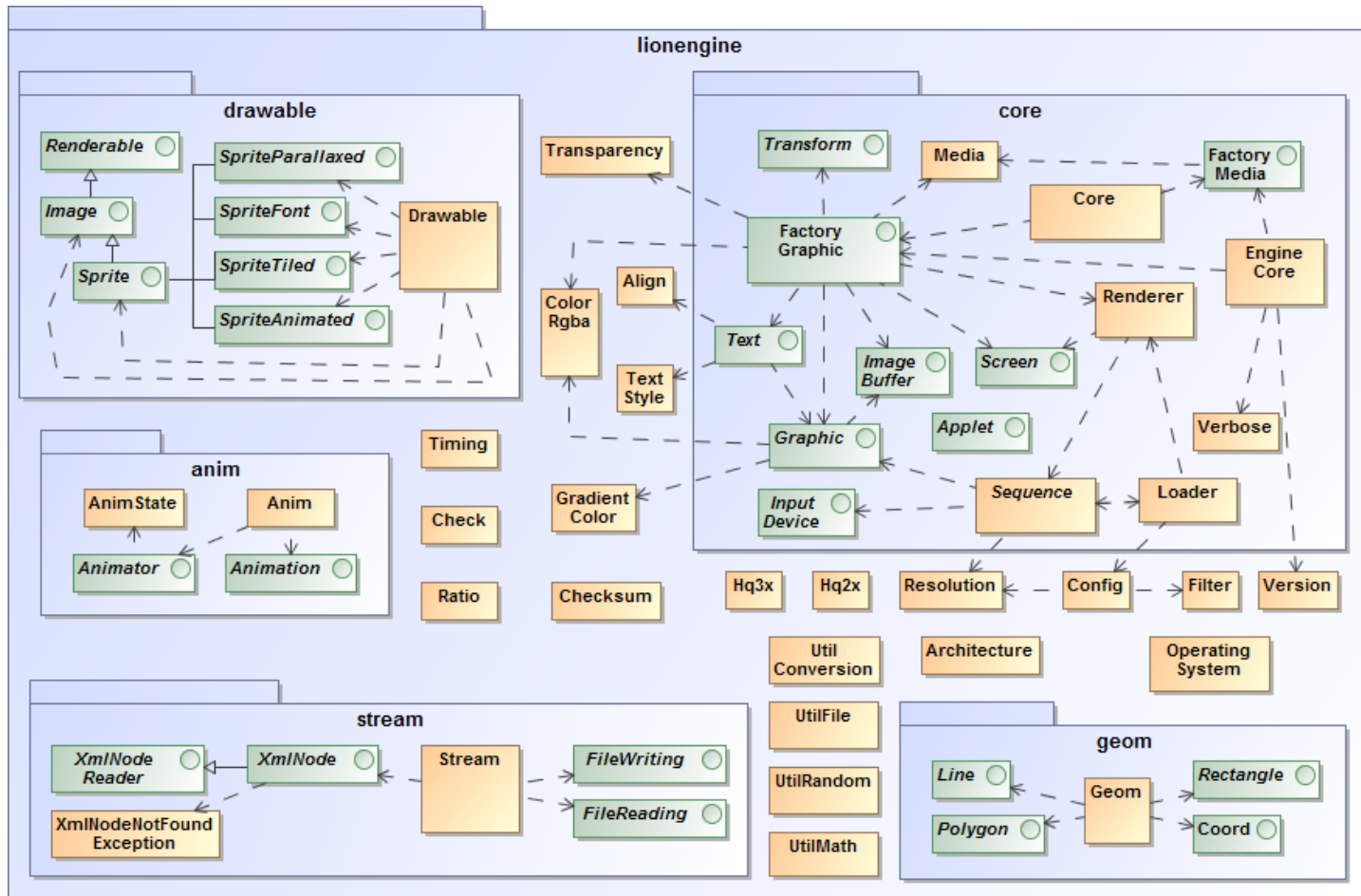
Module Core - Drawable

15

- **Renderable** (élément affichable simplement)
 - ▣ **Image** (surface non modifiable)
 - **Sprite** (surface modifiable)
 - **SpriteAnimated** (surface animée)
 - **SpriteTiled** (surface découpée en rectangles)
 - **SpriteParallaxed** (surface pour un effet 2.5D)
 - **SpriteFont** (police d'écriture depuis une image)

Module Core - UML

16



Plan

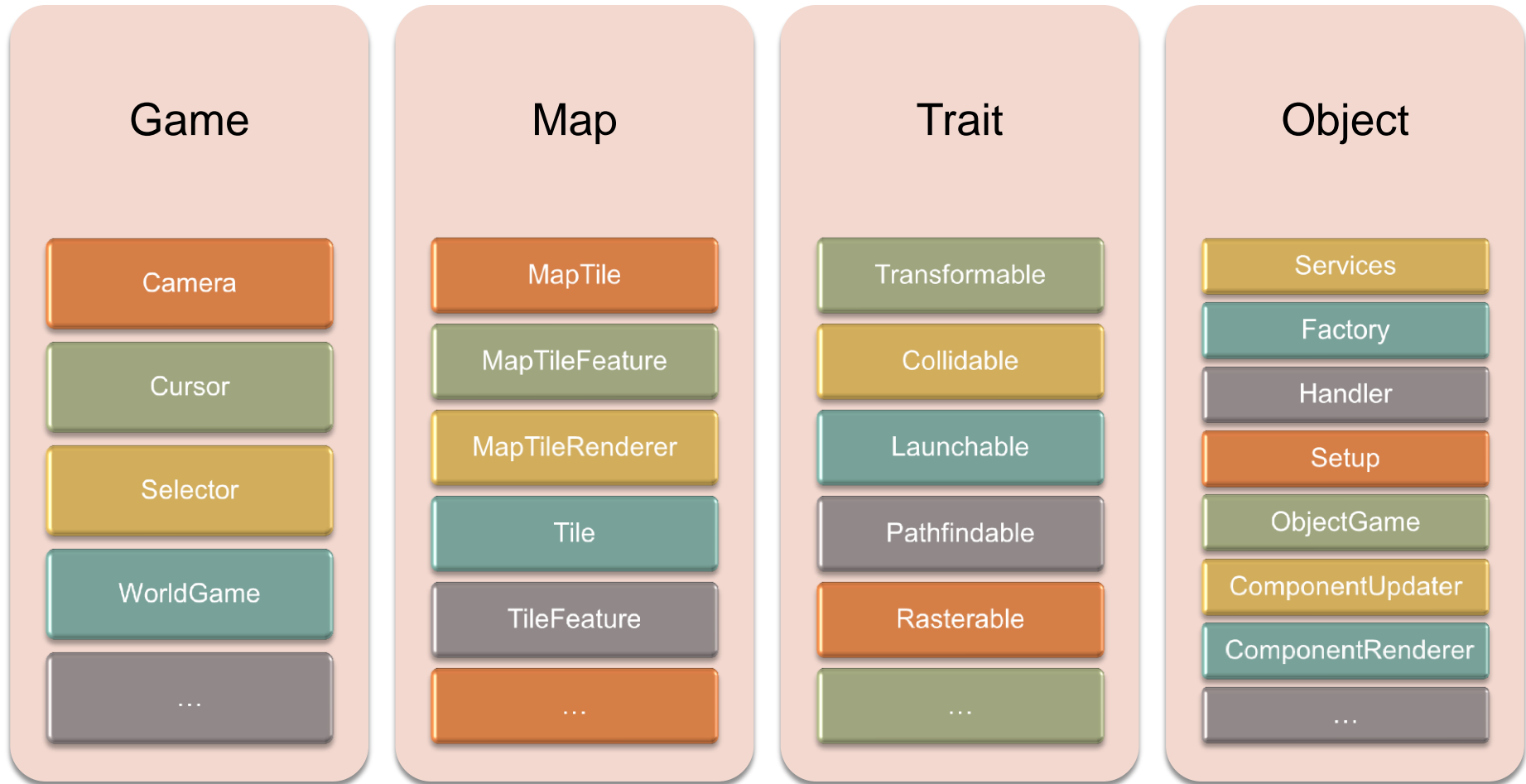
17

- Moteur
- Module Core
- **Module Game**
- Module Network
- Editeur



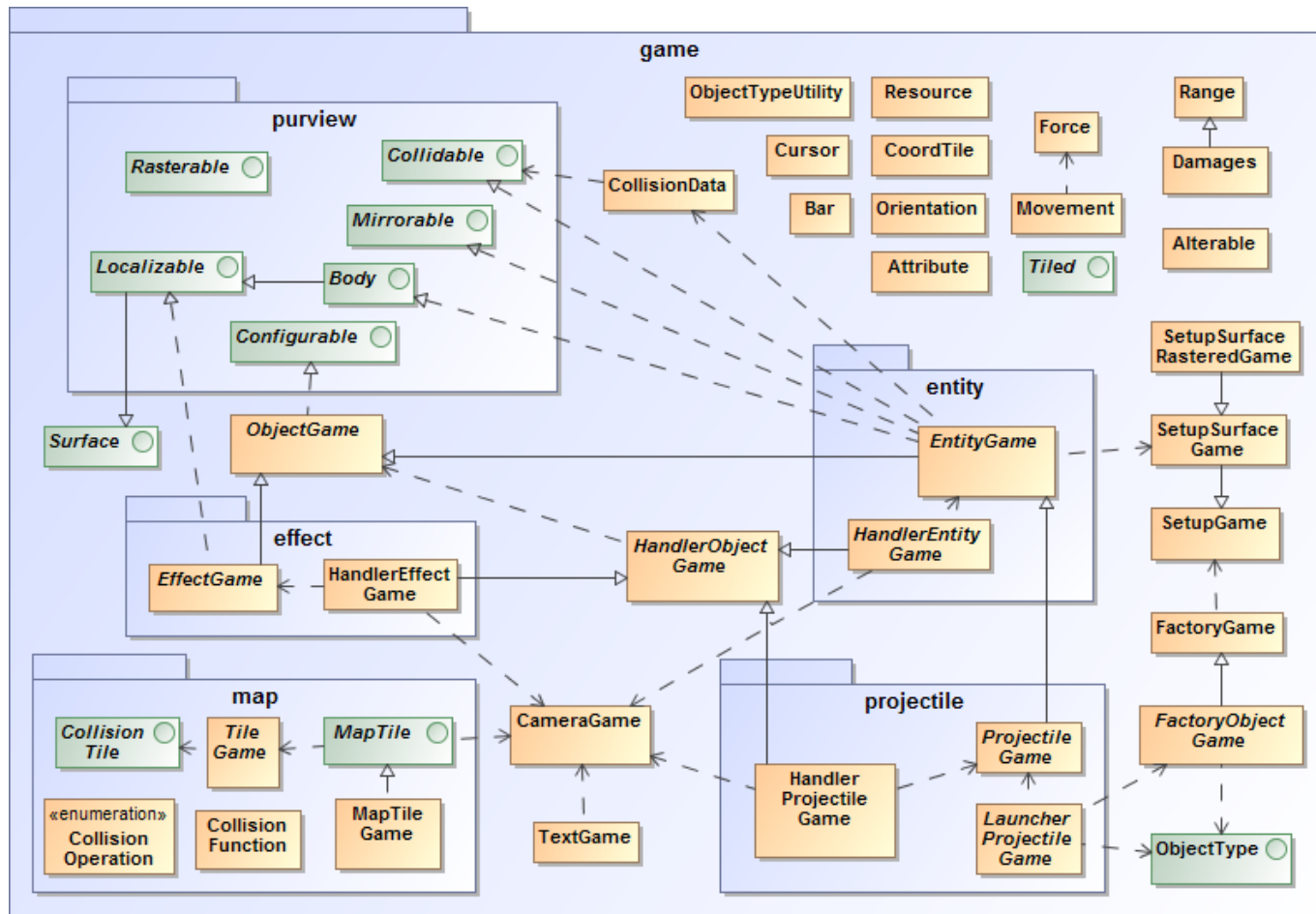
Module Game - Structure

18



Module Game - UML

19



Module Game - Game

20

- **Dans le package** : `com.b3dgs.lionengine.game`
- **Propose des types primaires**
 - ▣ **ObjectGame** (représente un objet de base)
 - ▣ **Camera** (vue du joueur, évoluant dans le jeu)
 - ▣ **Factory** (chargé de créer les objets)
 - ▣ **Handler** (gère une collection d'objet)
 - ▣ **WorldGame** (conteneur: handler, map, camera...)
 - ▣ ...

Module Game - Map

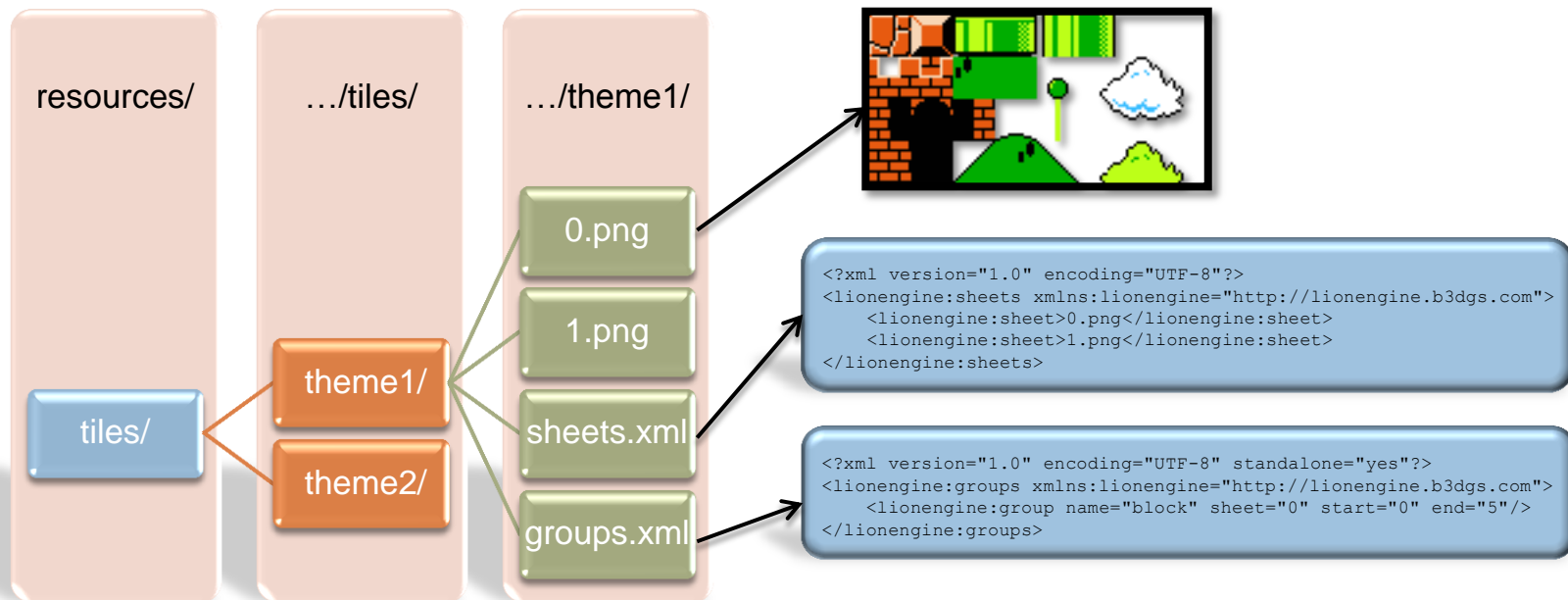
21

- Dans le package: `com.b3dgs.lionengine.game.map`
- Propose un type de map standard
 - ▣ MapTile (interface décrivant une map à base de tile)
 - MapTileGame (implémentation abstraite de base)
 - Chargement des tiles dans une image (SpriteTiled)
 - Import & export au format binaire
 - Associe les collisions aux tiles à partir d'un fichier externe
 - Génération d'une minimap représentant la map en pixel
 - TileGame (structure de base d'un tile)
 - Numéro de pattern (=N°image d'une tuile)
 - Numéro de tile (=N°tile dans la tuile)
 - Nom de la collision associée
 - Coordonnées sur la map (x, y)

Module Game - Map

22

- L'architecture impose une structure de rangement
 - ▣ Un dossier, placé dans le dossier des ressources, doit contenir un dossier par thème, eux même contenant la liste des tilesheets avec le fichier 'sheets.xml'
 - ▣ Sans ce fichier, toutes les images de type .png seront chargées



Module Game - Map

23

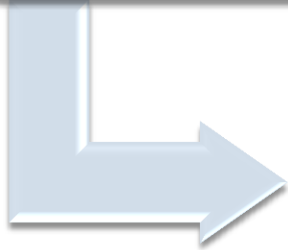
- La gestion des collisions est effectuée
 - ▣ En amont côté Tile
 - Récupère son type de collision depuis un fichier externe
 - Définit les points de collision en fonction d'une localisation
 - Dépend du type de collision du tile
 - ▣ En aval côté Entité
 - Teste quel est le premier tile traversé ayant une collision
 - Se place sur le point de collision du tile correspondant
 - Gère plusieurs collisions (sol, zones bloquantes...)

Module Game - Map

24

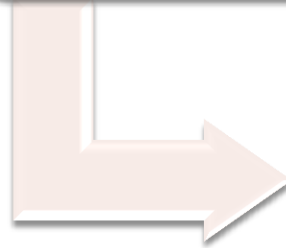
Check

- Recherche le premier tile ayant une collision suite à une intersection



Get

- Récupère le tile concerné, et recherche le point de collision associé



Apply

- L'entité se place sur le point de collision

Module Game - Trait

25

- **Dans le package:** `com.b3dgs.lionengine.game.trait`

- **Décrit des capacités supportées par des objets**
 - ▣ **Collidable** (gérant l'aspect collision de deux objets)
 - ▣ **Mirrorable** (permet d'avoir son symétrique)
 - ▣ **Transformable** (gère le placement d'un objet)
 - ▣ **Rasterable** (permet de gérer l'effet raster bar)
 - ▣ **Body** (représente un objet soumis à la gravité)
 - ▣ ...

Module Game - Utility

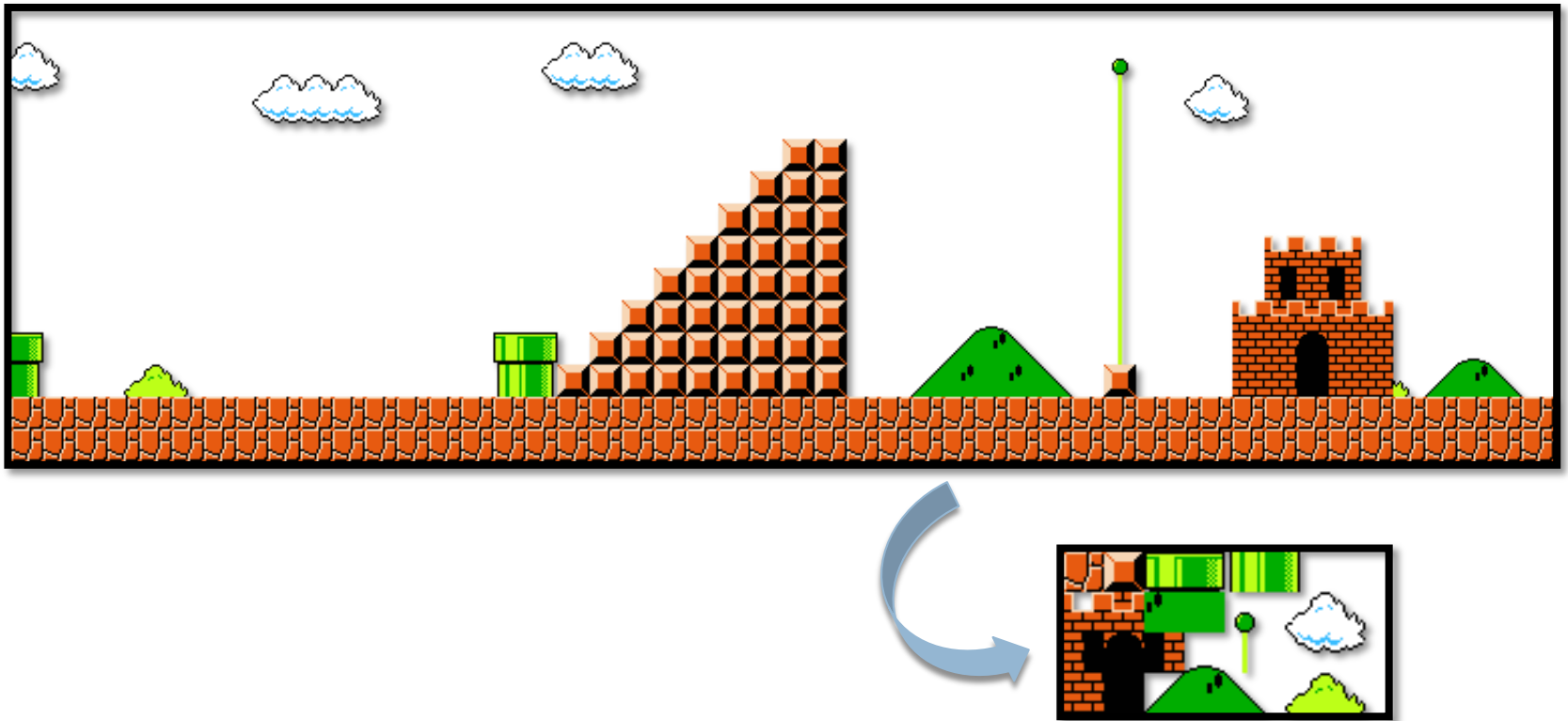
26

- Dans le package: `com.b3dgs.lionengine.game.map`
- Conversion d'un « levelrip », en un format de données compatibles MapTile
 - ▣ Charge un levelrip (image représentant un niveau entier)
 - ▣ Convertit au format MapTile
 - ▣ Sauvegarde au format binaire
 - ▣ Supporte le multithreading (meilleures performances)
- Extracteur de tile à partir d'un levelrip
 - ▣ Découpe les tiles uniques d'un levelrip et les sauvegarde dans une image en tilesheet

Module Game - Utility

27

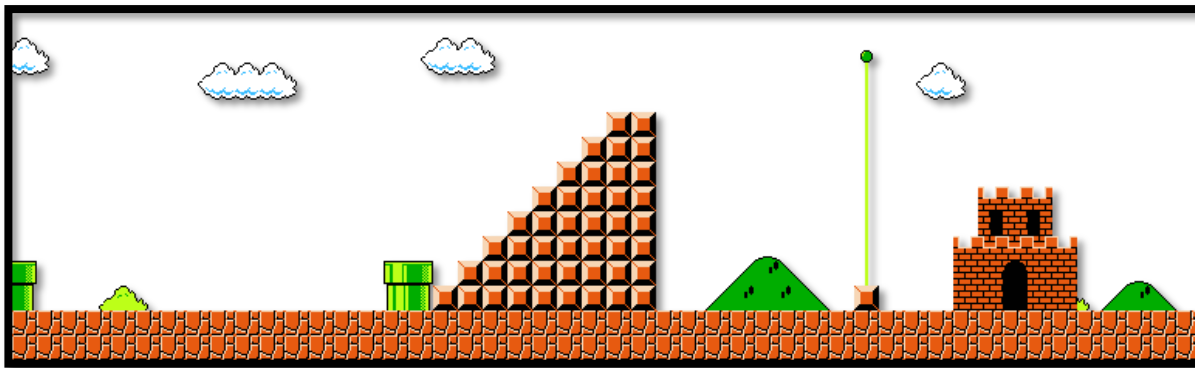
- *'TileExtractor'* en action:



Module Game - Utility

28

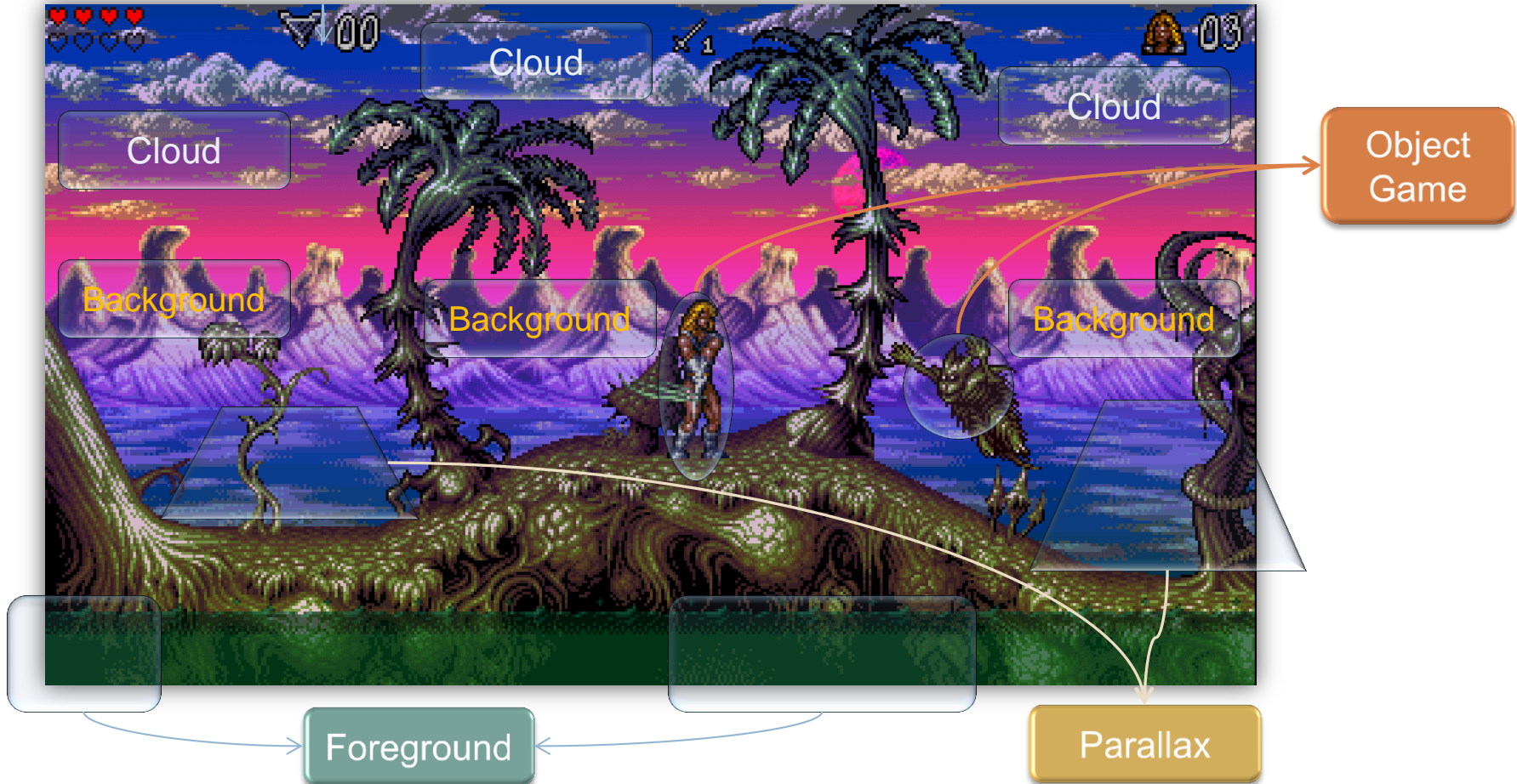
- *'LevelRipConverter'* en action:



MapTile
(data of
tiles)

Module Game - Exemple

29



Module Game - Exemple

30



Plan

31

- Moteur
- Module Core
- Module Game
- **Module Network**
- Editeur



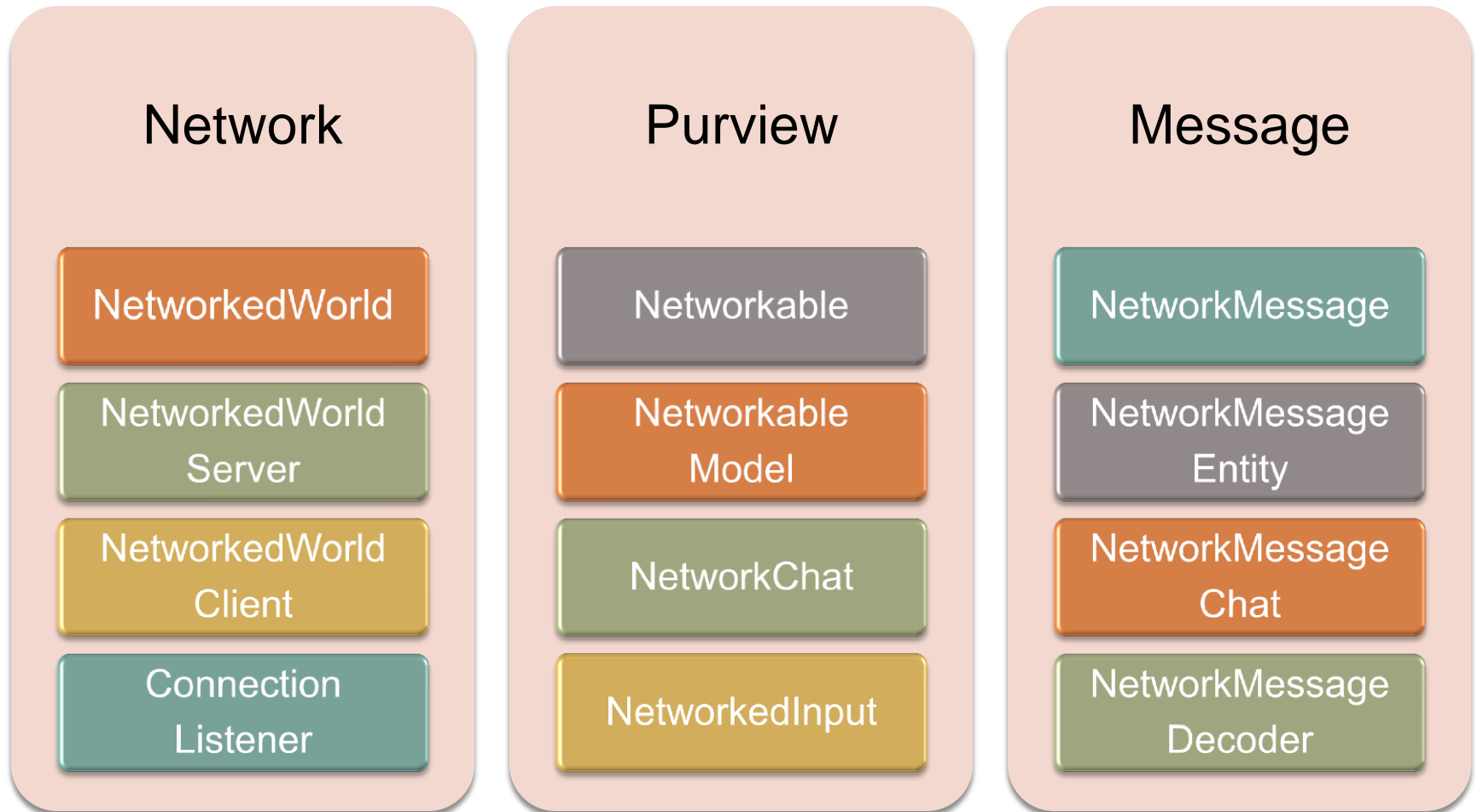
Module Network

32

- Module dédié aux jeux en réseau
 - ▣ Propose des types de base
 - Networkable
 - NetworkMessage
 - NetworkMessageDecoder
 - NetworkedWorldServer
 - NetworkedWorldClient
 - NetworkChat

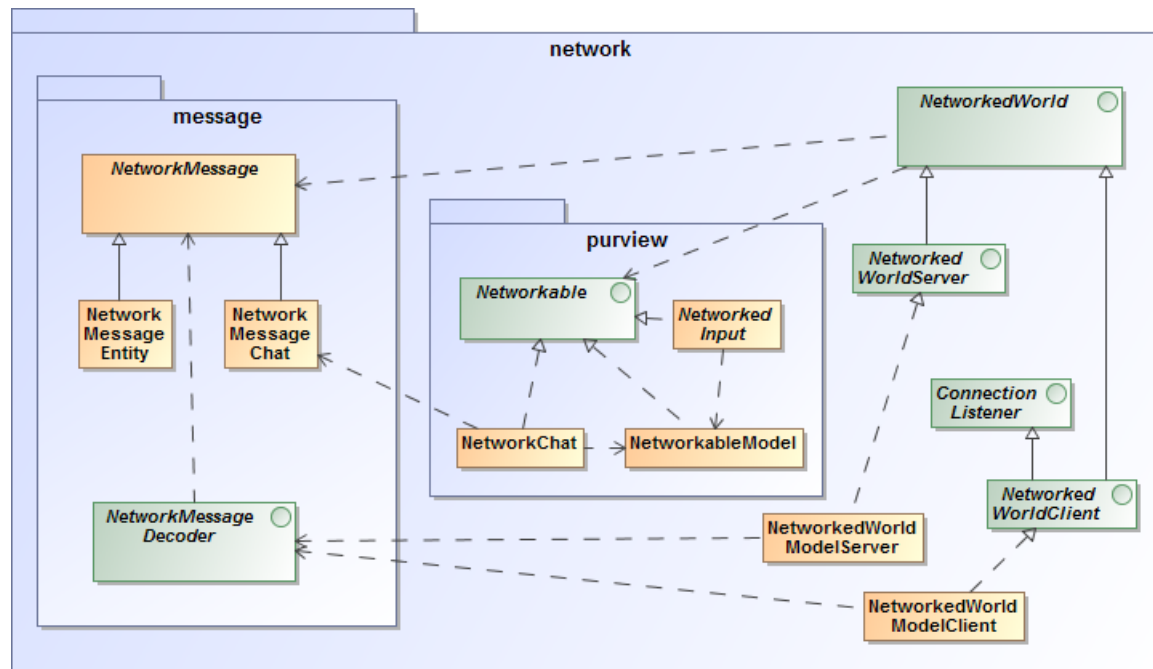
Module Network - Structure

33



Module Network - UML

34



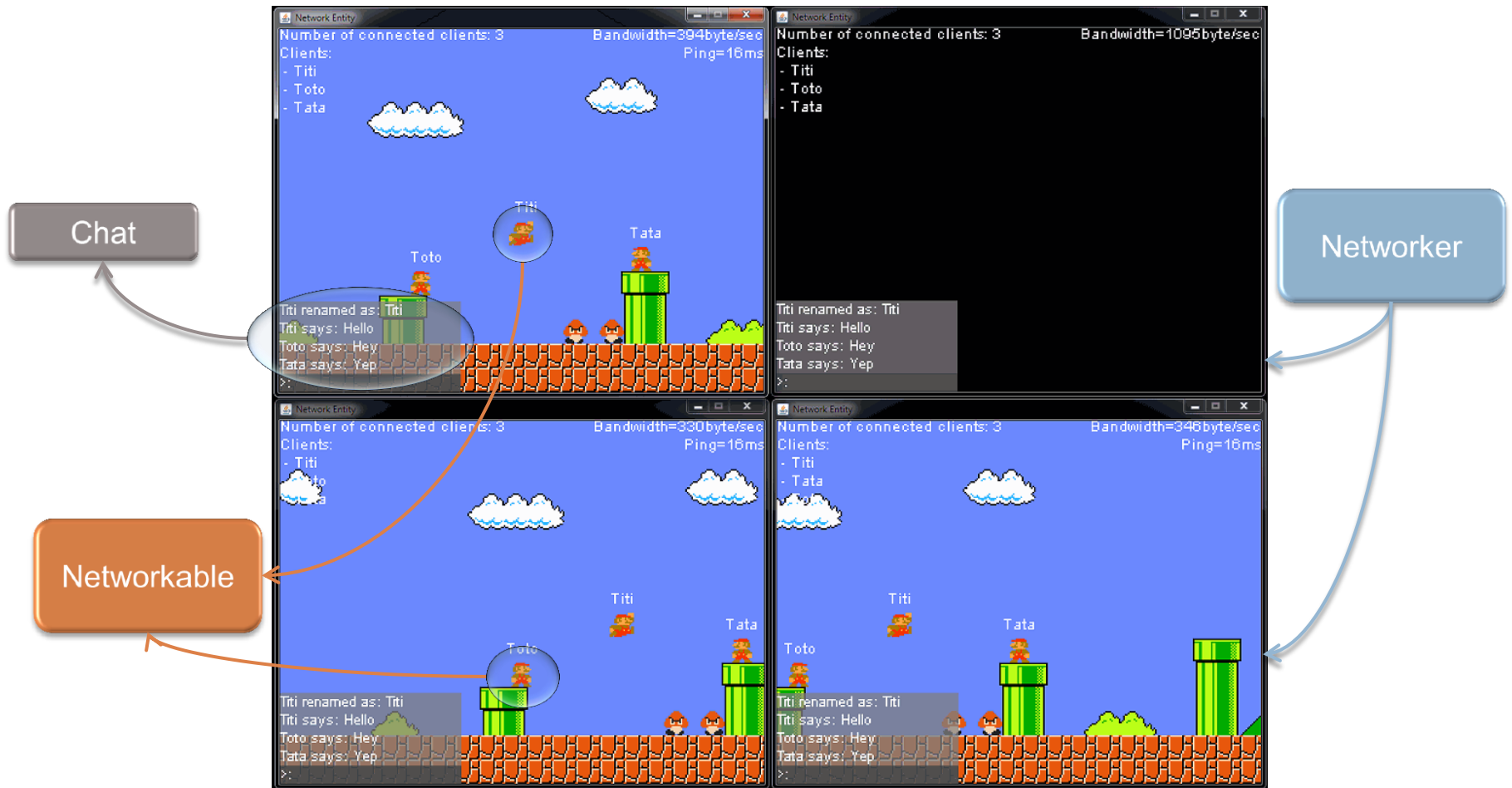
Module Network - Package

35

- Dans le package: `com.b3dgs.lionengine.network`
- Propose des types spécialisés « network »
 - ▣ `NetworkMessage` (message standard)
 - ▣ `NetworkMessageDecoder` (décode un message)
 - ▣ `Networkable` (support pour le réseau)
 - ▣ `NetworkedWorldServer` (monde côté serveur)
 - ▣ `NetworkedWorldClient` (monde côté client)
 - ▣ `NetworkChat` (chat standard)


Module Network - Exemple

36



Plan

37

- Moteur
- Module Core
- Module Game
- Module Network
- **Editeur** 

Editeur

38

- Repose sur Eclipse RCP 4
- Utilisable tel quel
- Extensible pour gérer ses spécificités
- Requiert le module lionengine-core-swt

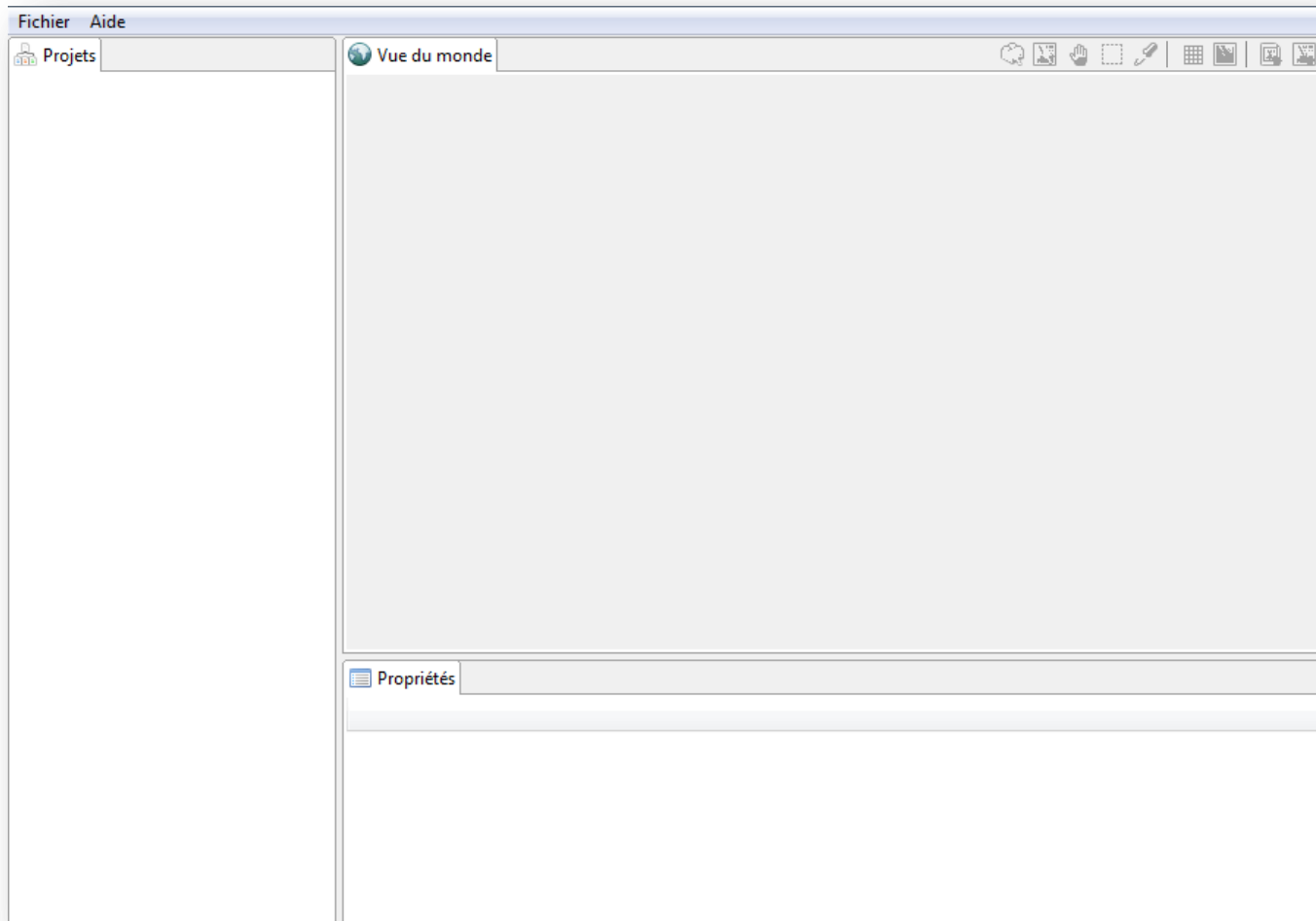
Editeur - Fonctionnalités

39

- Importer un projet utilisant le LionEngine
- Gestion complètes des tilesheets
 - ▣ Génération depuis un level rip
 - ▣ Importer un niveau depuis des tile sheets et un level rip
 - ▣ Gérer des groupes, formules, collisions...
- Gestion des objets
 - ▣ Editeur d'animations, collisions, attributs spécifiques...
- Gestion du monde
 - ▣ Placement / suppression d'objets, navigation...

Editeur - Vide

40



Byron 3D Games Studio - LionEngine 8.0.0

Editeur - Exemple

41

