19/12/2015

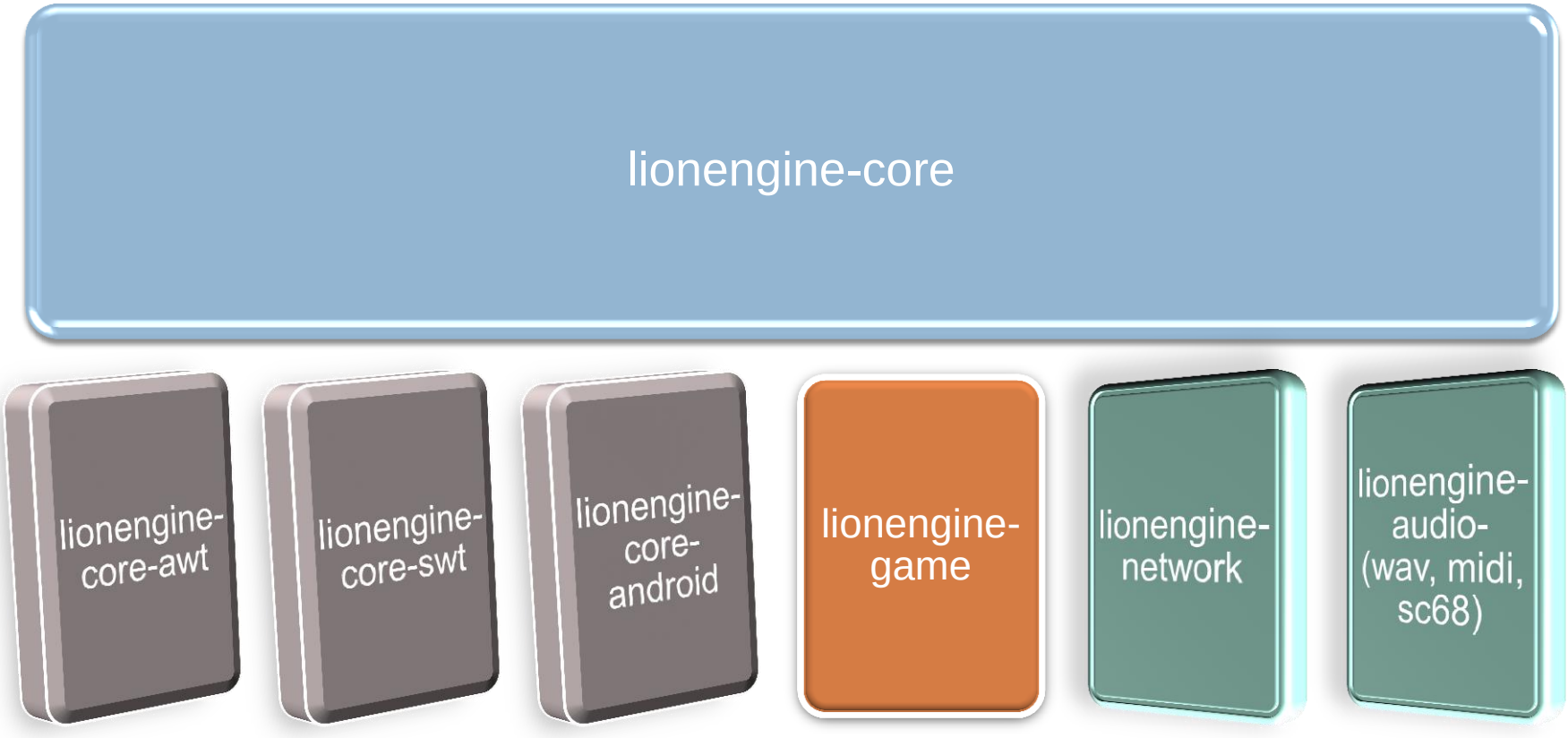
Un moteur de jeu 2D en Java

Plan

- **Moteur**

- Module Core
- Module Game
- Module Network
- Editeur

Moteur


lionengine-core
lionengine-core-awt
lionengine-core-swt
lionengine-core-android
lionengine-game
lionengine-network
lionengine-audio-(wav, midi, sc68)

Moteur

- 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, scène…)

Moteur

- API « haut niveau »
  - Abstraction de premier niveau
    - Classes de base orientées jeux-vidéo généraux
    - Routines 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 (A*)
      - …

Moteur - Modules

- 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

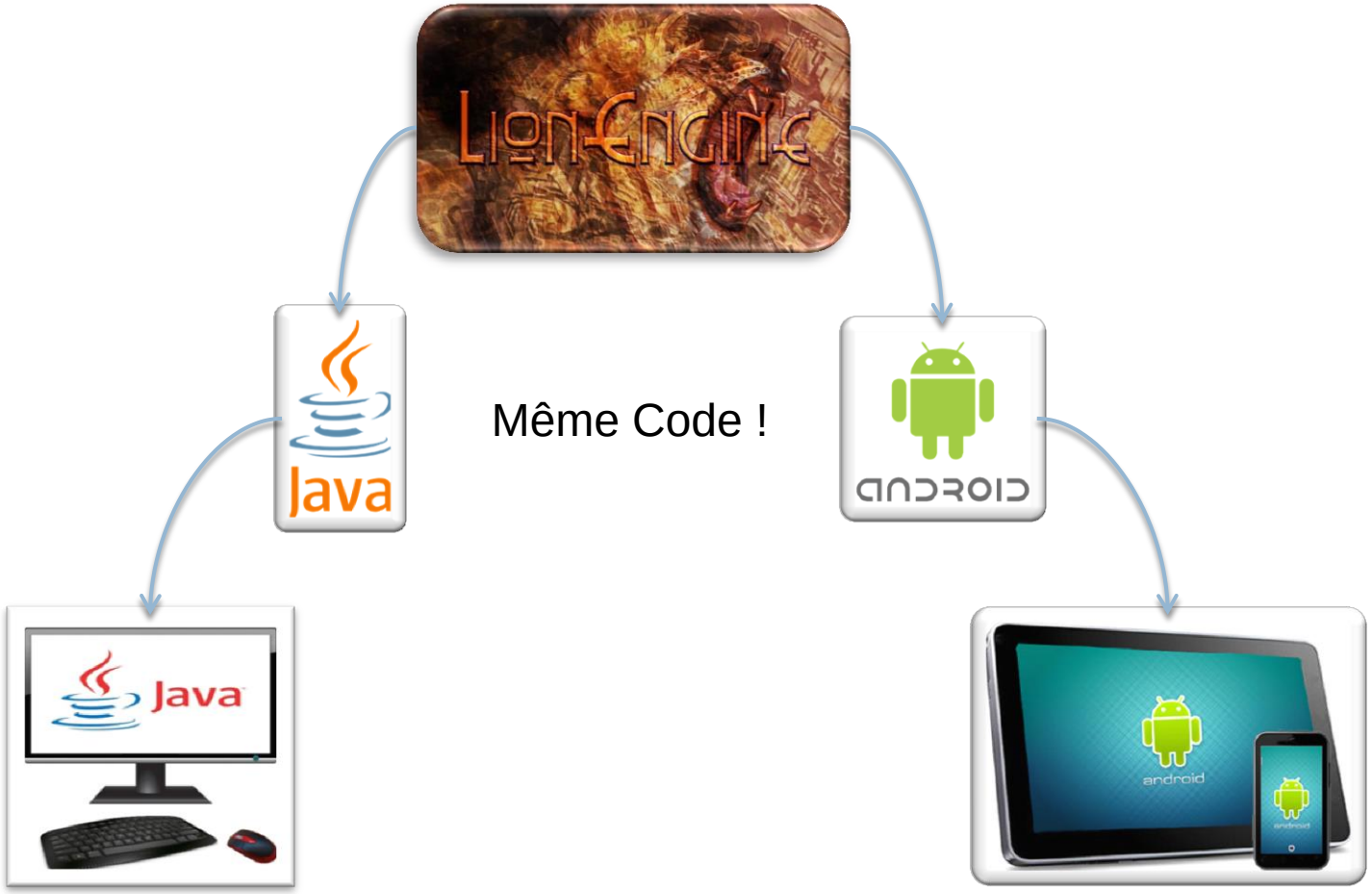
- Chaque module
  - Dépend du module principal `(lionengine-core)`
  - Propose une base abstraite `(architecture de base)`
  - Est redéfinissable selon les besoins, en tout point
  - Est compatible avec d'autres modules
  - Respecte la même structure que la partie centrale

Moteur - AWT, SWT, Android

- Le moteur ne dépend pas d'implémentation bas niveau directe

- 3 modules au choix
  - lionengine-core-awt
    - PC, fenêtré, plein écran, et applet en utilisant Java2D
  - lionengine-core-swt
    - PC, fenêtré et plein écran en utilisant SWT (3.5.1)
  - lionengine-core-android
    - Smartphones et tablettes, avec Android 1.5 au minimum

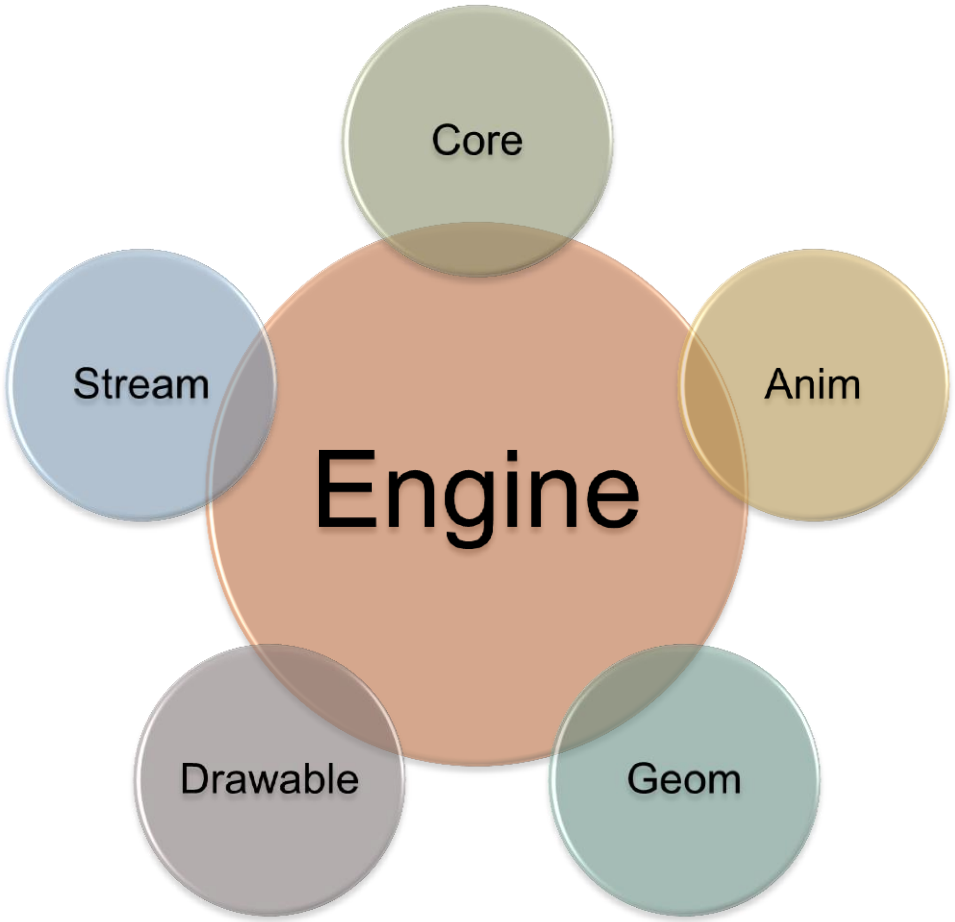
Moteur - Utilisation



Plan

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

Module Core



Module Core

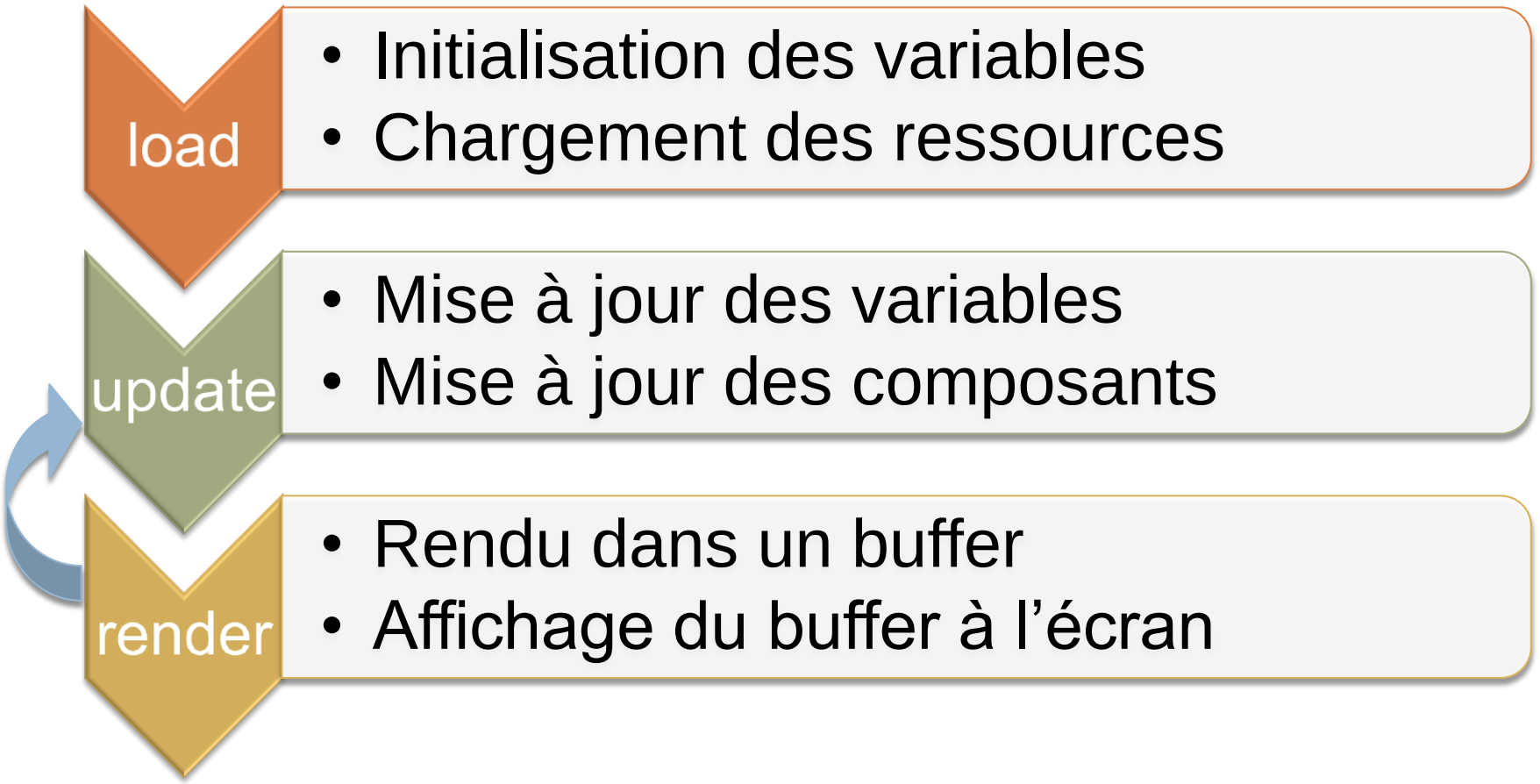
- Situé à partir du package : `com.b3dgs.lionengine`
- Principaux packages / classes
  - anim *(Animator, Animation, AnimState)*
  - core *(Engine, Config, Graphic, Sequence…)*
  - drawable *(Image, Sprite, SpriteTiled, SpriteFont…)*
  - stream *(FileReading, FileWriting, XmlFactory, XmlNode)*
  - geom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

Module Core - Sequence

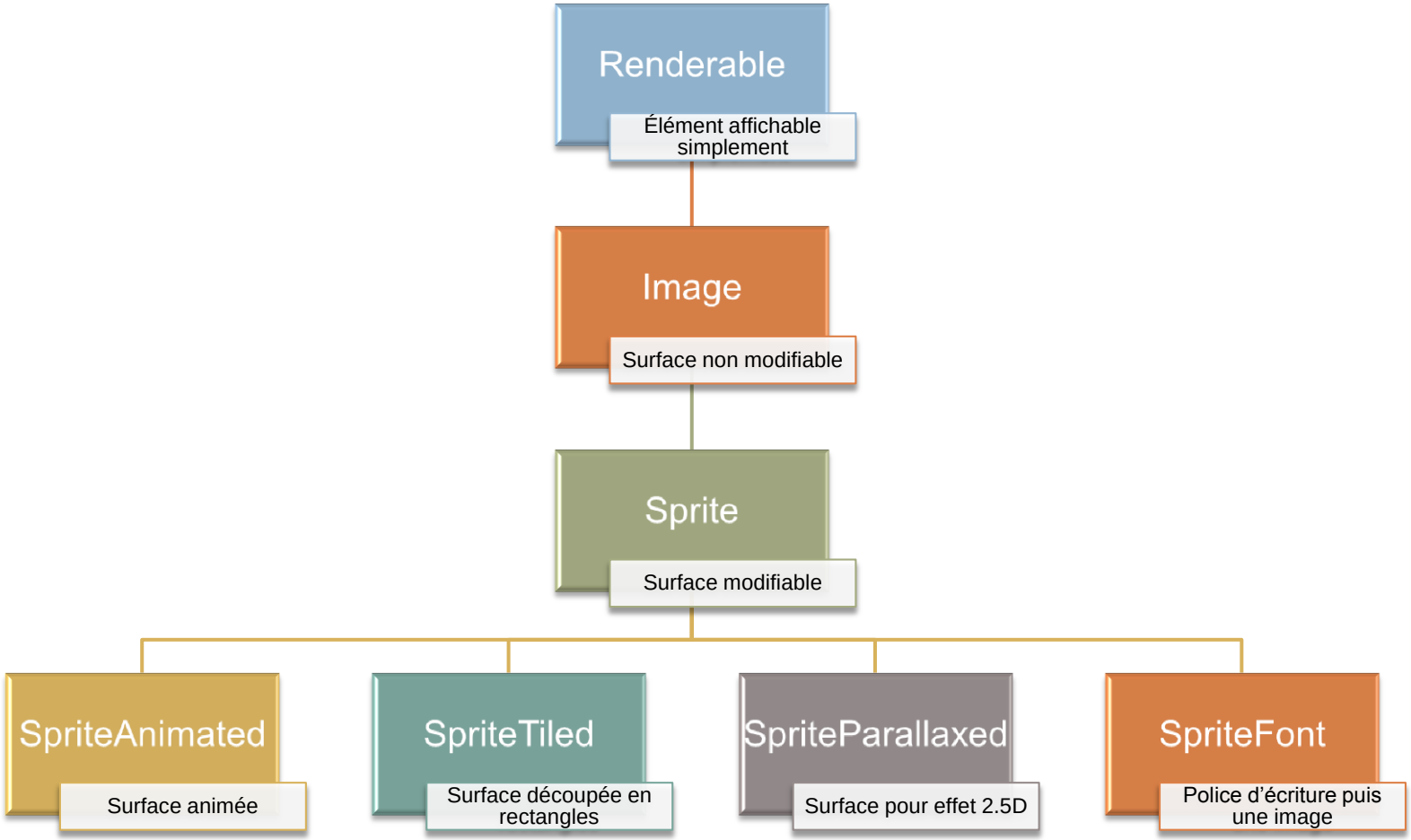
- Squelette de base
  - `load();`
  - `update(double extrp);`
  - `render(Graphic g);`
  - `onTerminate(); // Optionnel`

- Gestion du nombre d'images par seconde
- Gestion de l'extrapolation (`'machine independant'`)
- Modes d'affichage: `plein écran, fenêtré, applet`

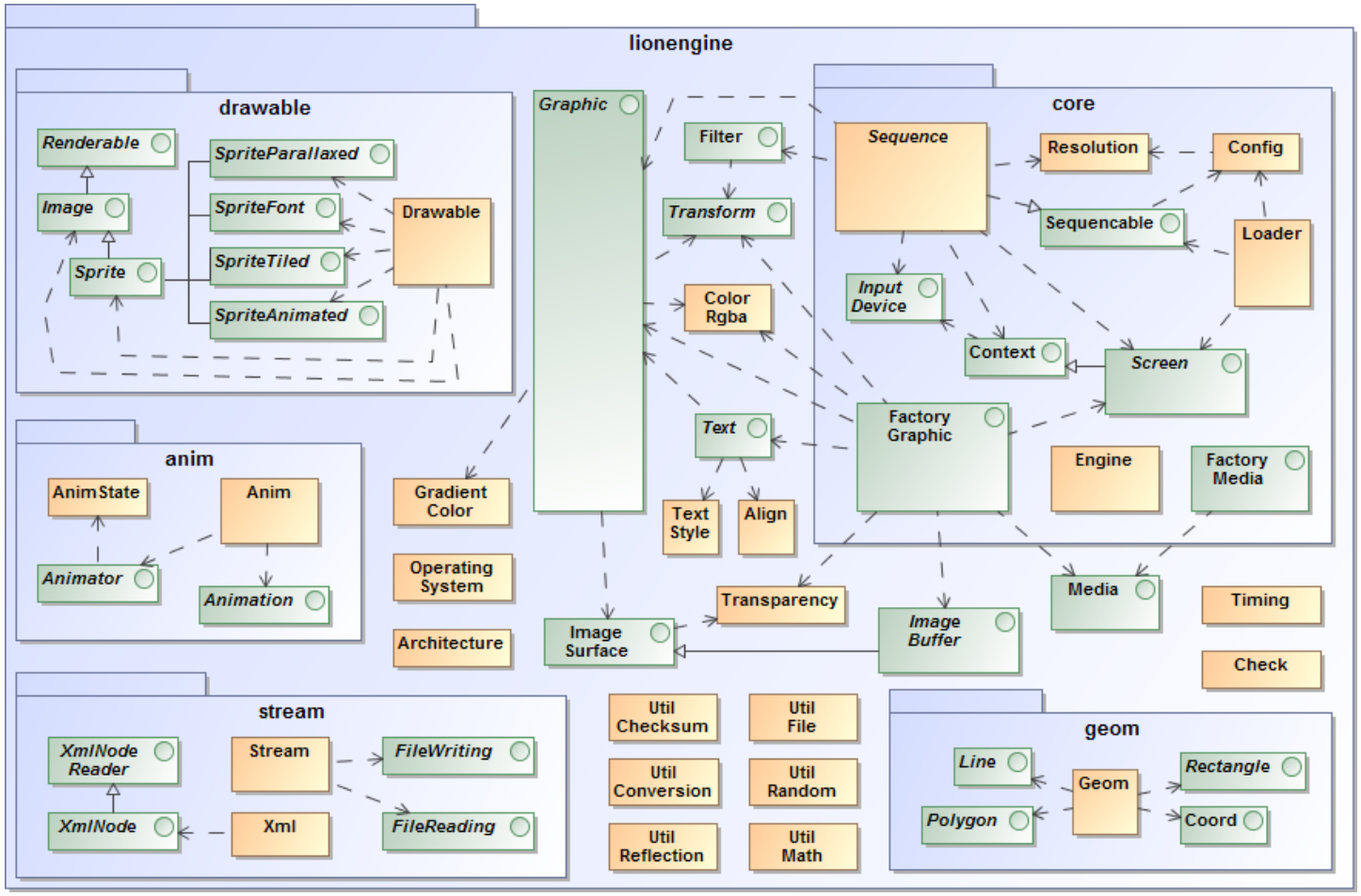
Module Core - Sequence



Module Core - Drawable



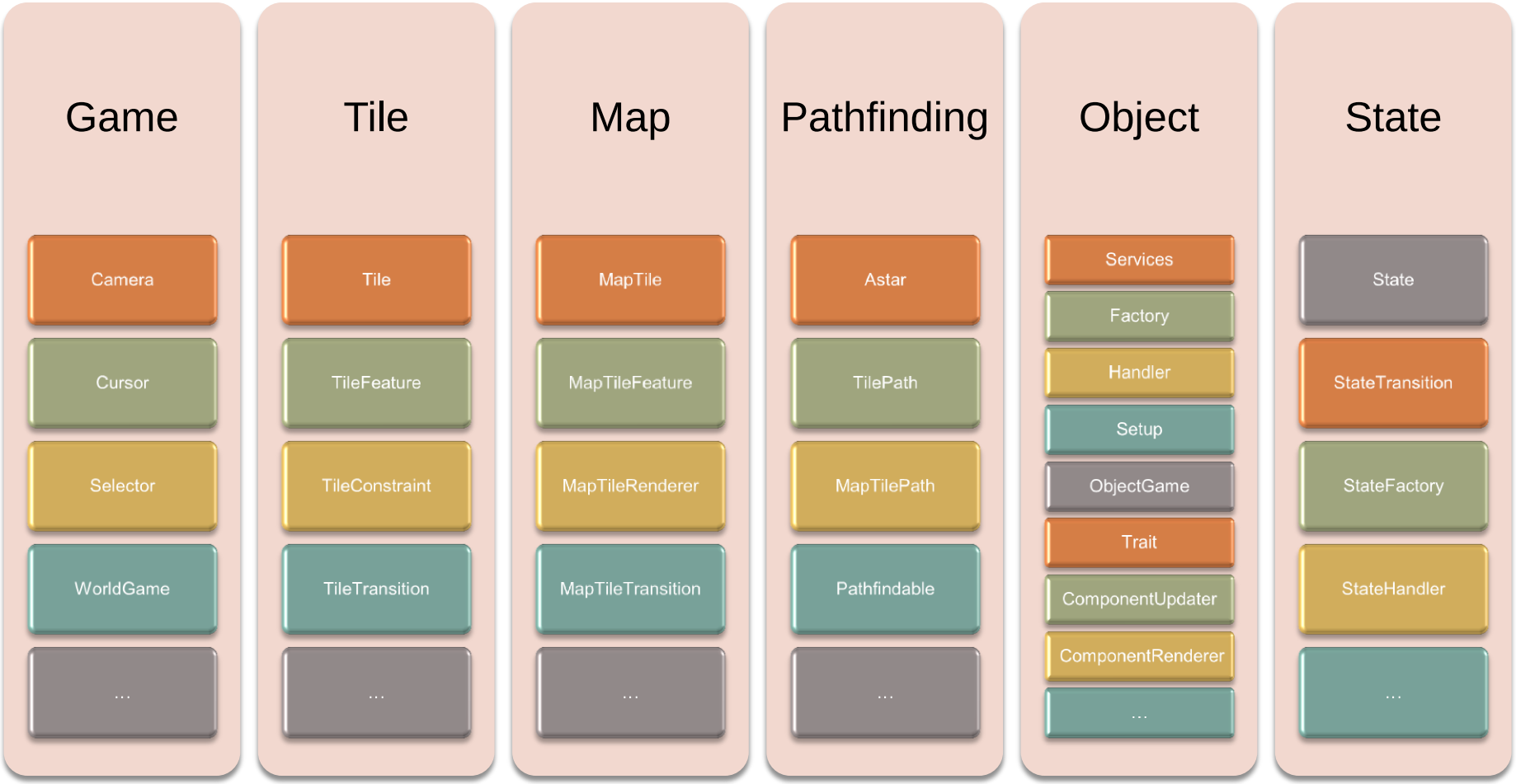
Module Core - UML



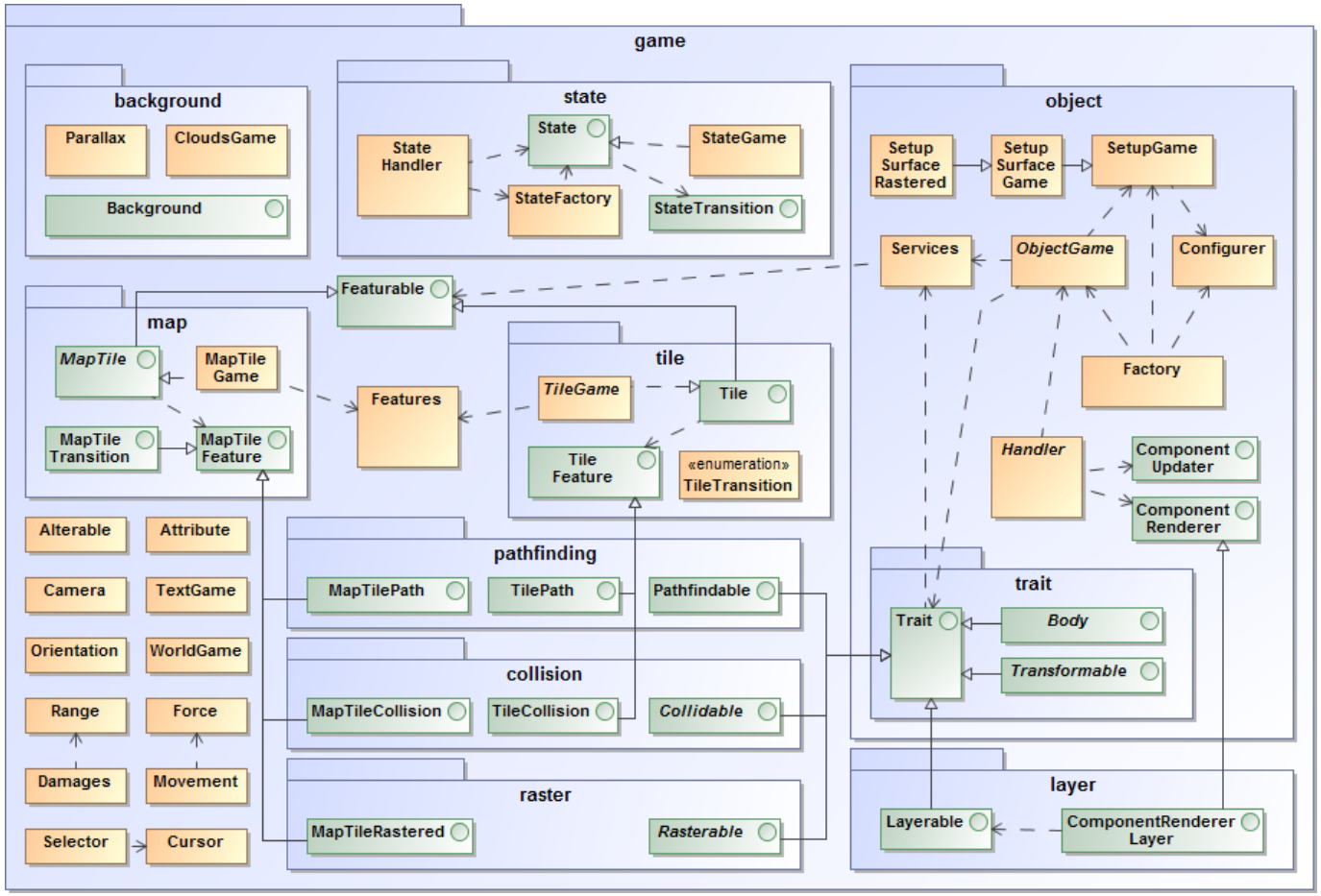
Plan

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

Module Game - Structure



Module Game - UML



Module Game - Game

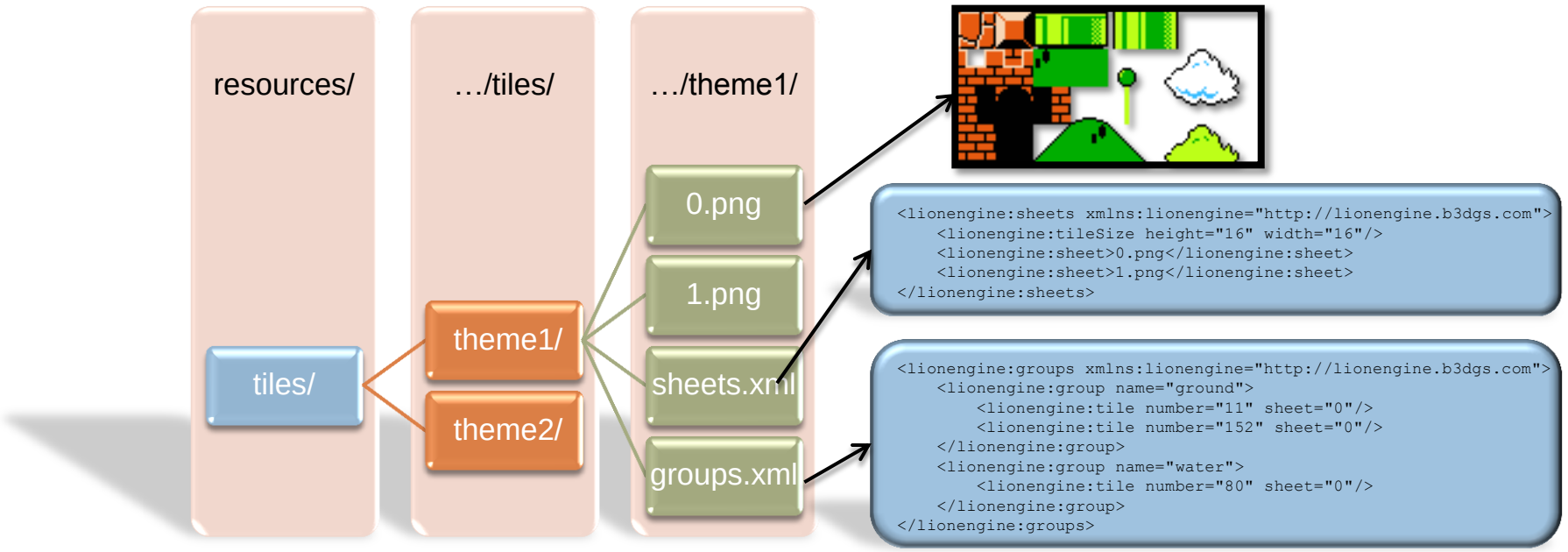
- Dans le package : `com.b3dgs.lionengine.game`
- Propose des types primaires
  - ObjectGame `(représente un objet de base)`
  - Camera `(vue du joueur, évoluant dans le jeu)`
  - Factory `(chargé de créer les objets)`
  - Handler `(gère une collection d'objets)`
  - WorldGame `(conteneur : handler,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

Module Game - Tile & Map

- 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 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
    - 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 groupe associée
      - Coordonnées sur la map (x, y)

Module Game - Map

- 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, rien ne sera chargé

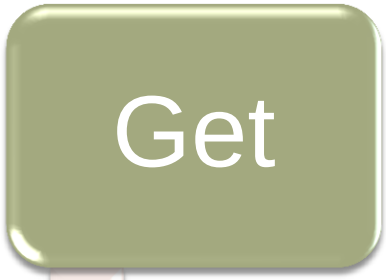


Module Game - Map

- 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

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



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

- L'entité se place sur le point de collision

Module Game - Trait

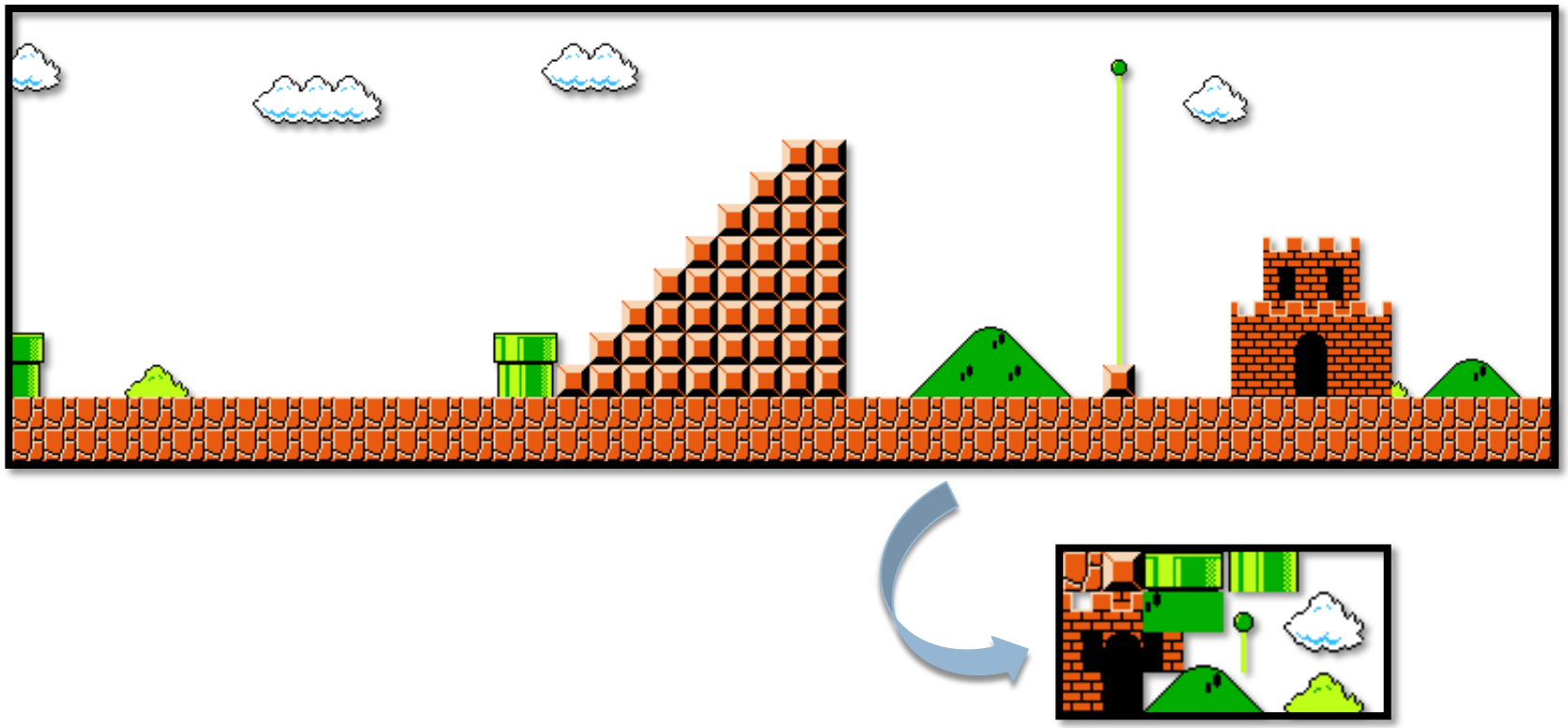
- Dans le package : `com.b3dgs.lionengine.game.trait`
- Décrit des capacités supportées par des objets
  - Collidable `(gérant la collision de deux objets)`
  - Mirrorable `(permet d'avoir le symétrique)`
  - Transformable `(gère le placement d'un objet)`
  - Rasterable `(permet de gérer l'effet raster bar)`
  - Body `(représente un objet soumis à la gravité)`
  - …

Module Game - Utility

- 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
- Extracteur de tile à partir d'un levelrip
  - Découpe les tiles uniques d'un levelrip et les sauvegarde dans une image en tilesheet

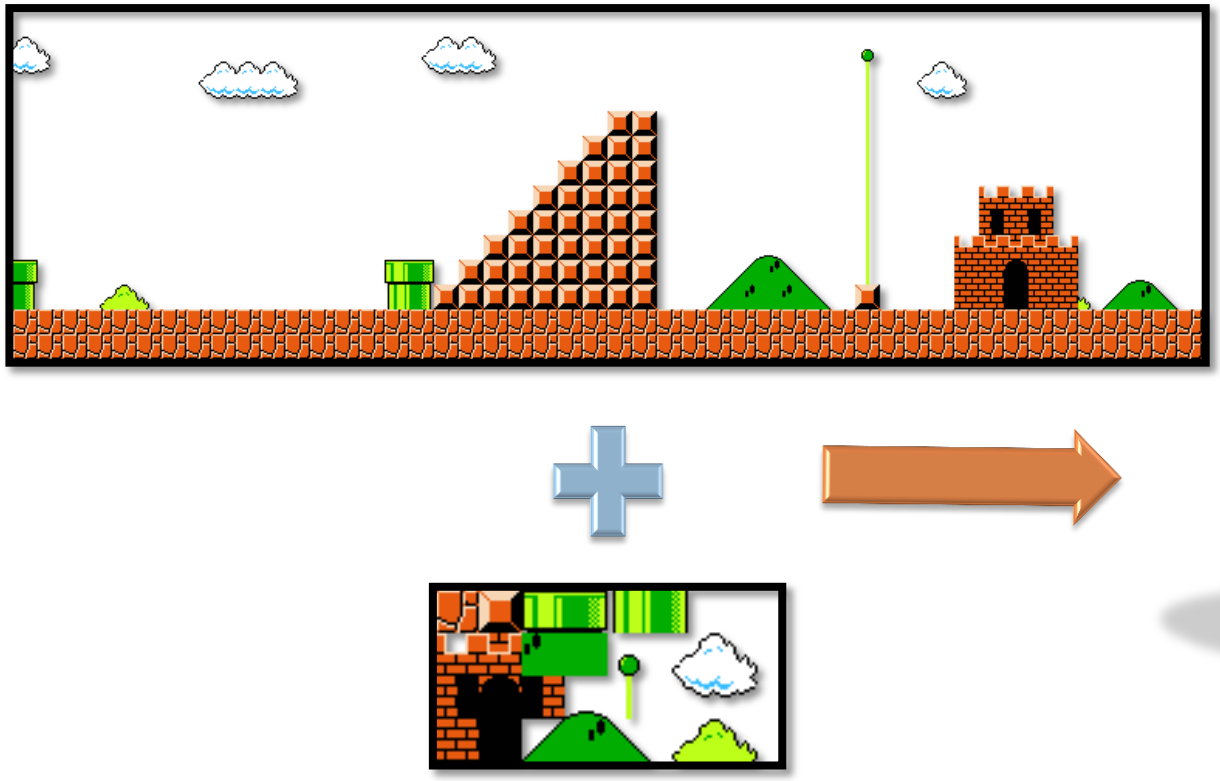
Module Game - Utility

- '*TileExtractor*' en action:

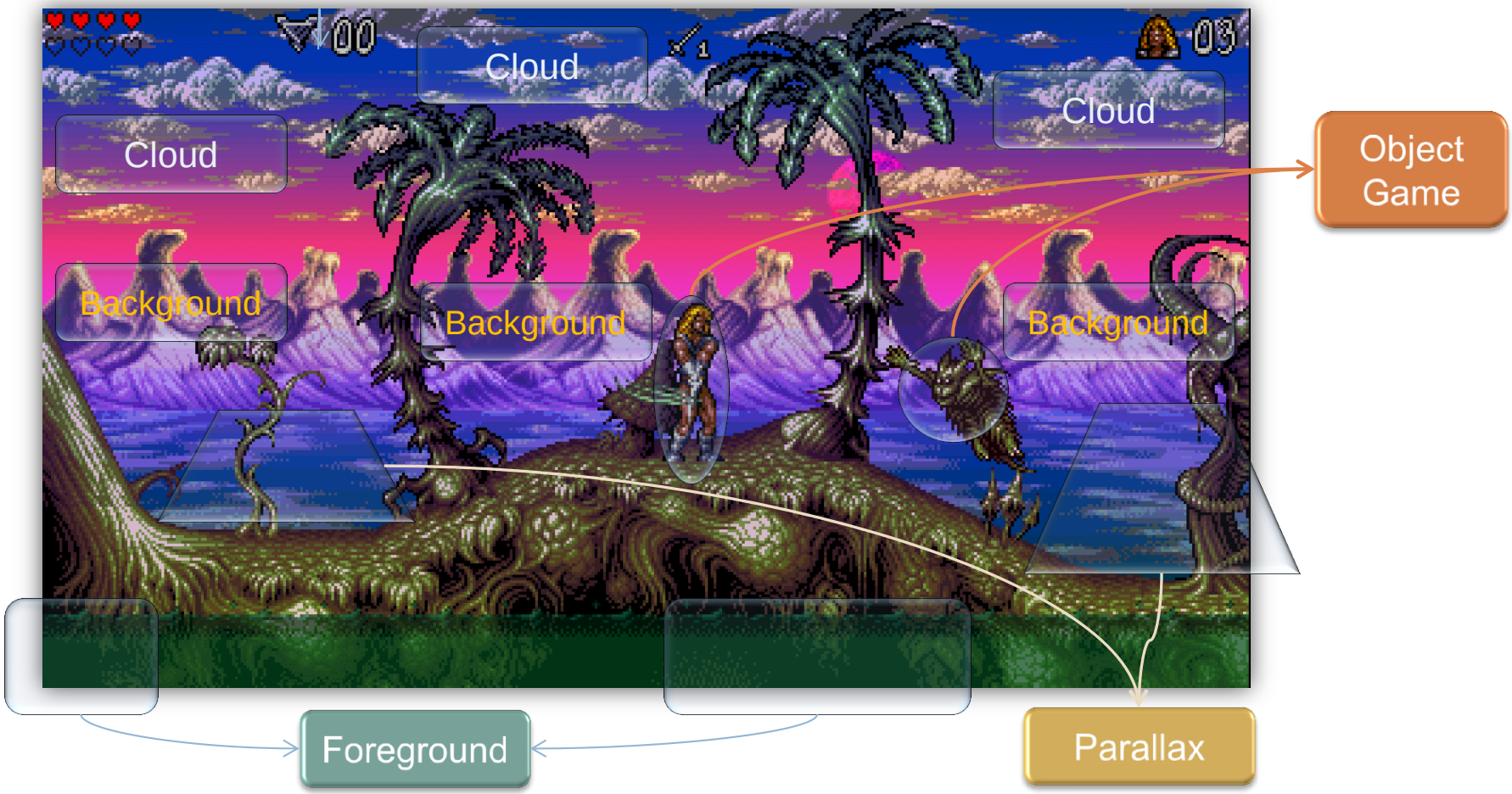


Module Game - Utility

- '*LevelRipConverter*' en action:



Module Game - Exemple



Module Game - Exemple

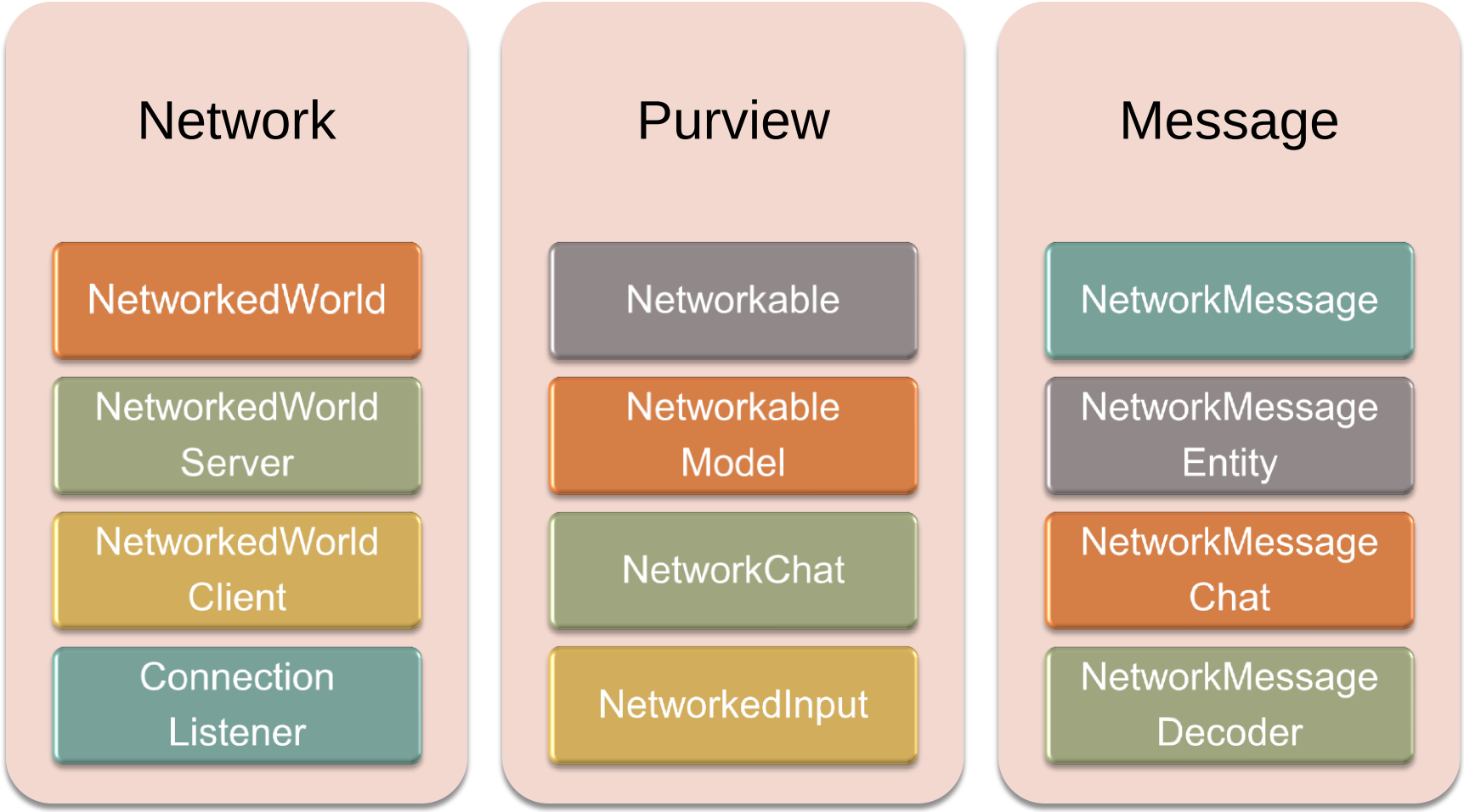
Plan

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

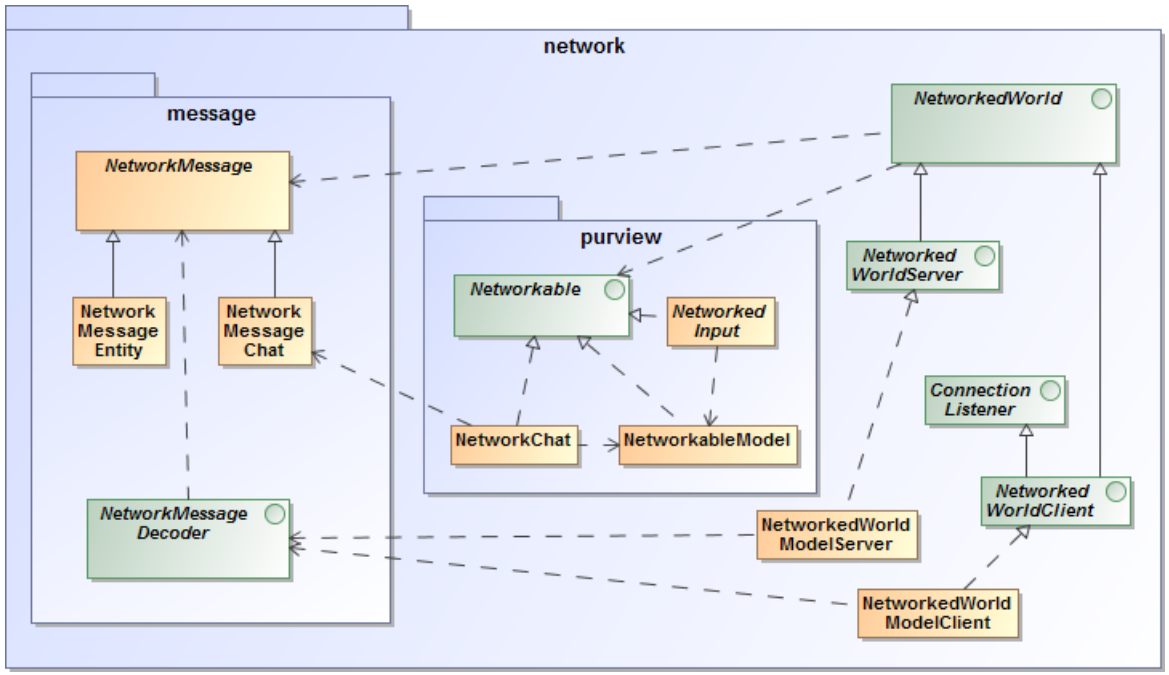
Module Network

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

Module Network - Structure



Module Network - UML



Module Network - Package

- 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

Plan

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

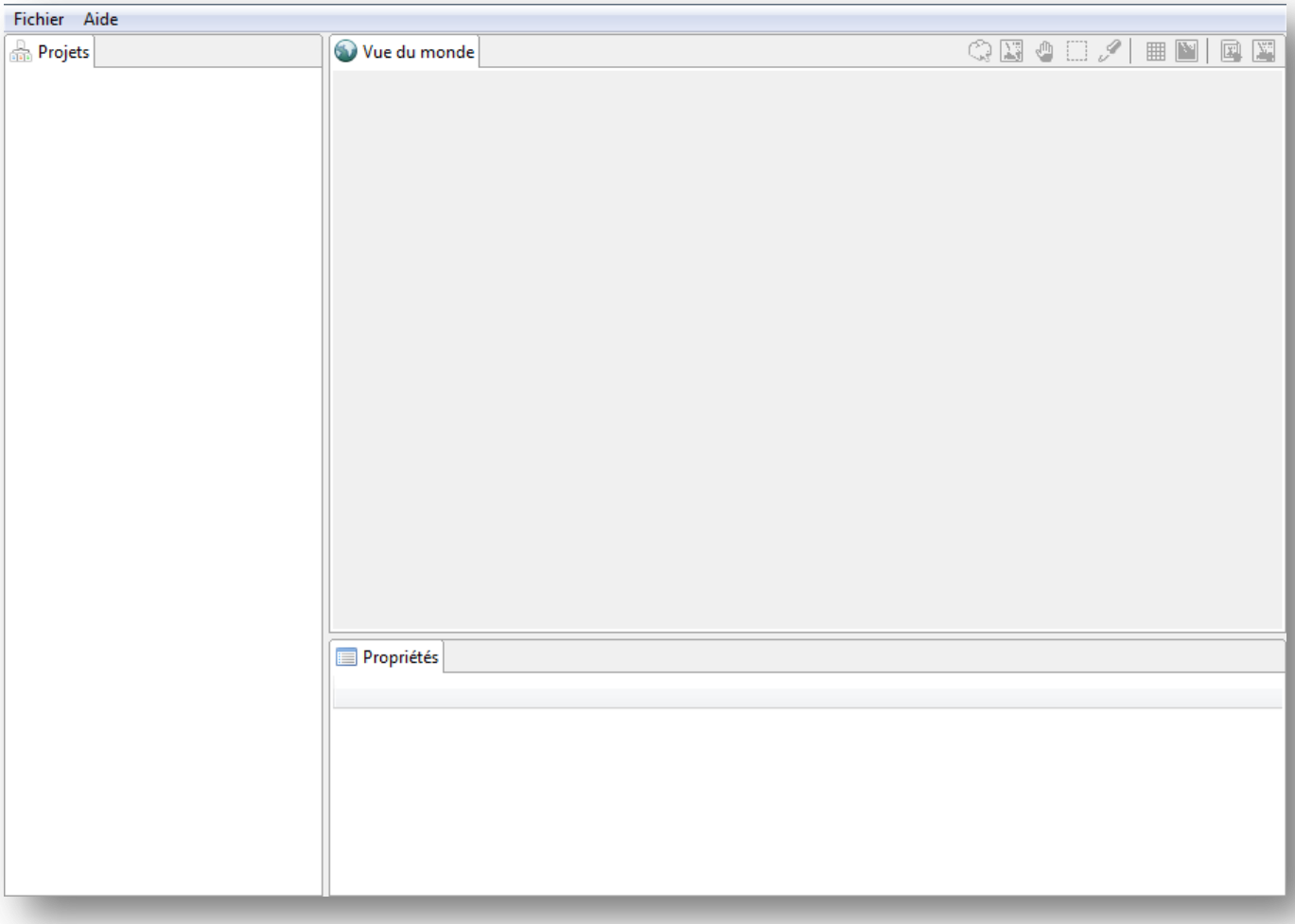
Editeur

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

Editeur - Fonctionnalités

- 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 specifiques…
- Gestion du monde
  - Placement / suppression d'objets, navigation…

Editeur - Vide



Editeur - Exemple