**Monkey** Gaming Framework

Features

* AS3 platform
  + General application workflow managing
  + RSL managing
  + Resource managing
  + Object initialization and disposal management
  + Logging and Error Managing
  + Event Management (Signals)
* Core gaming
  + Object State Management core features
  + Screening Managing
  + Sound Managing
  + Quality and performance (Q&P) managing
  + Layering and GUI managing core features
  + Camera managing core features
  + Input Management (Mouse, Keyboard)
* Physics managing
  + Extensible Abstract Physics engine managing
  + Out-of-the-box Box2D engine integration
  + PHML markup language
* Framework capabilities
  + Plugin architecture
* Compiler & IDE
  + Extensions to mxmlc compiler
  + Eclipse integration

Design guidelines

* IOC (Inversion of control) Pattern   
  Monkey framework will provide a general application workflow that will be customizable and extensible via predefined extension points such as plugins and other patterns.
* Dependency injection  
  Monkey framework will be fully modular, allowing the developer to use only the features he needs for his projects and those created by himself, allowing a general contributed evolution of the framework.
* Markup languages for common tasks  
  Monkey framework will promote the use of markup languages to define many aspects of the workflow customization, allowing a clearer source code.
* Convention over configuration
* KISS (Keep it simple and stupid) - Simple coding
* Lightweight coding, keep performance in mind
* Composition over inheritance.
* Document everything , comment everything

Inversion of control pattern: Game workflow

Game Workflow

Screen Workflow

* Retrieve screen resources
* Initialize screen model
* Initialize screen view
* Start Frame workflow

OnFrame Workflow

* Update model
* Process Input queue
* Physics engine step & CCD (Continuous collision detection) (optional)
* Q&P calculations and updates
* Update view
* Execute bindings