

[GameLab]

The First step towards a Component and Data-Oriented Game

February 7, 2022

Objectives: The objective is to implement a small component and data-oriented Asteroids game in *libGDX*. By the end of Project [GameLab], you will be familiar with basic data-oriented game programming in *libGDX*. Based on the implemented game we will discuss the pros and cons of the implementations. Last, the goal is to identify parts of the game that can be refactored into components using *provided* and *required* interfaces and a data-oriented design.

Classwork:

- Document the provided `IGamePluginService`, `IEntityProcessorService` and `IPostEntityProcessorService` interfaces using JavaDoc. Consider the pre- and post-conditions for each method signature.
- Implement the `Player` and `Enemy` as a separate project using the provided `IGamePluginService`, `IEntityProcessorService` and `IPostEntityProcessorService` interfaces. See provided example [\[AsteroidsEntityFramework\]](#) in GitHub.
- Implement randomly moving Asteroids using the `IGamePluginService` and `IEntityProcessorService` interfaces.
- Implement a simple collision detection system based on *Pythagoras* and the provided `IPostEntityProcessorService` interface. Ships that collide with asteroids should be destroyed. When fired upon Asteroids should split into two smaller Asteroids and when small enough they should be destroyed. Last, the player ship and enemy-ships should be destroyed when hit by each others bullets a certain number of times.
- Identify missing components.
- Specify at contract level (operation contracts in unified process), the required and provided interfaces based on identified components. That is, specify pre and post-conditions for each operation of the component.

Resources:

- Read [\[Data-Oriented Game Design\]](#).
- Read [\[Simple Collision Detection\]](#).
- For the interested reader, investigate [\[Box2d\]](#).