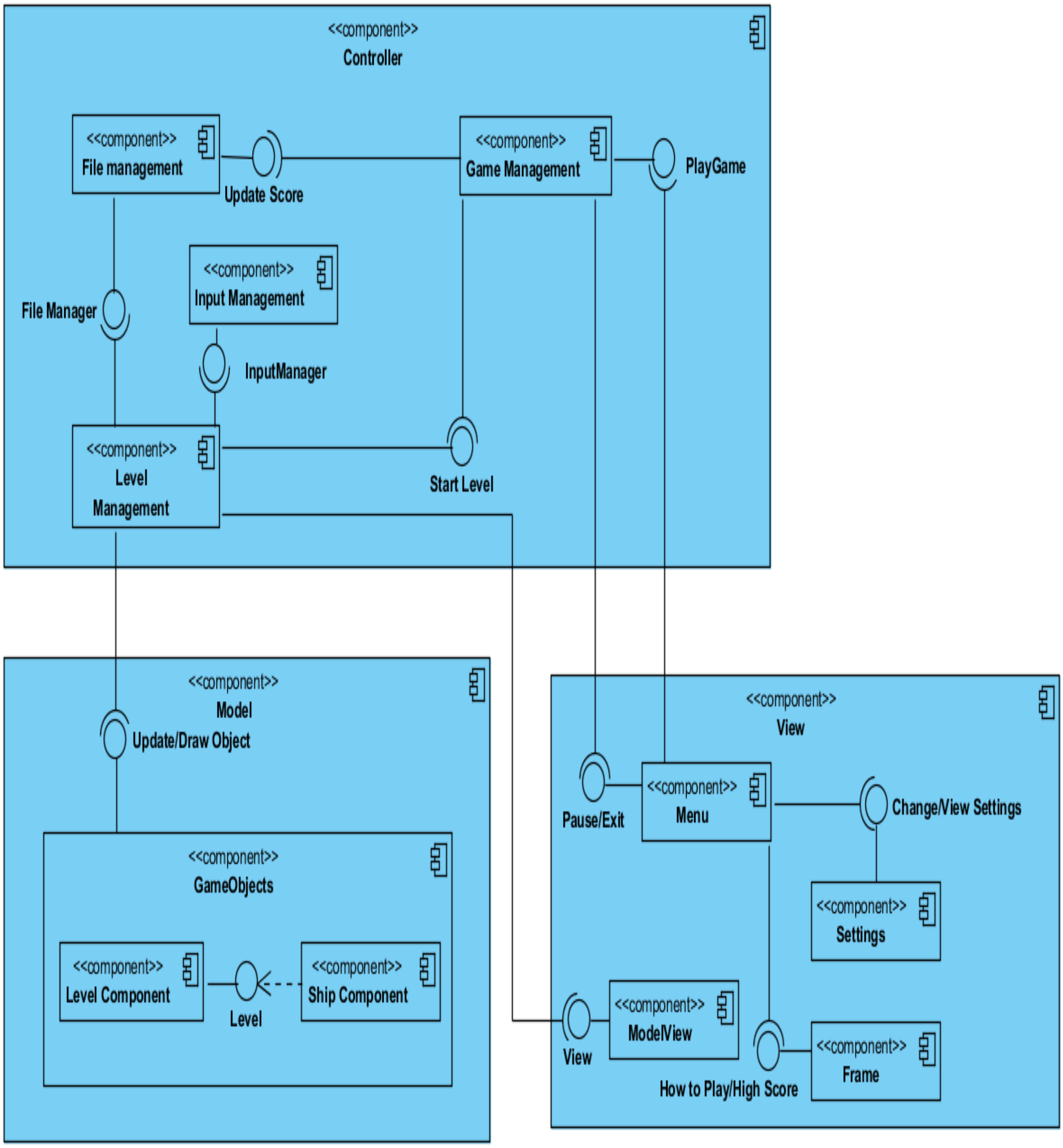
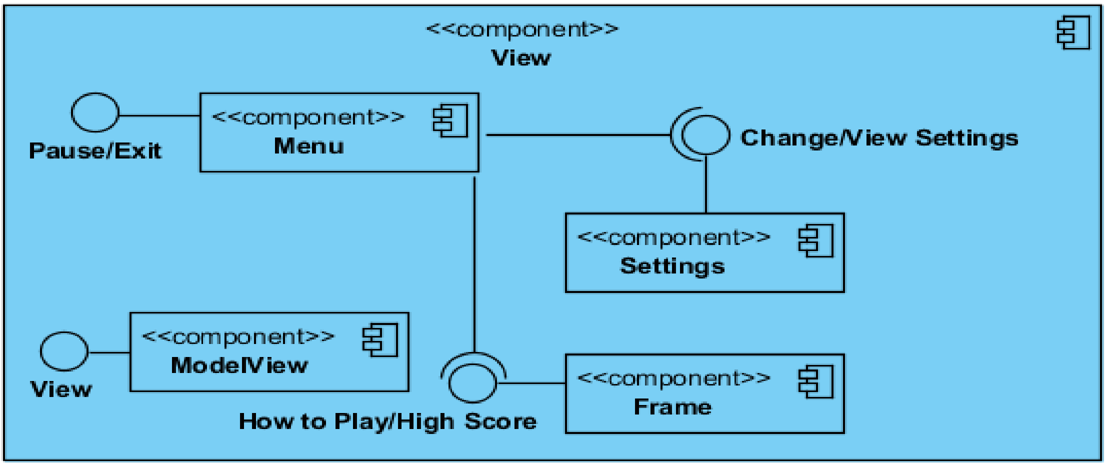
Subsystem Decomposition



3. Subsystem Service

3.1. View Subsystem



View Subsystem is providing user interface for “Space Invaders” game. Four major components of this subsystem are:

1. **Menu Component**
2. **ModelView Component**
3. **Frame Component**
4. **Settings Component**

Before starting the game, **Menu Component** provides a menu to the user. Menu component has a calss for Pause Menu and it contains Exit option. Furthermore, Menu Component can invoke and instantiate other components such as frame component and settings component. Menu component is the interface that provides different functionalities for the interface.

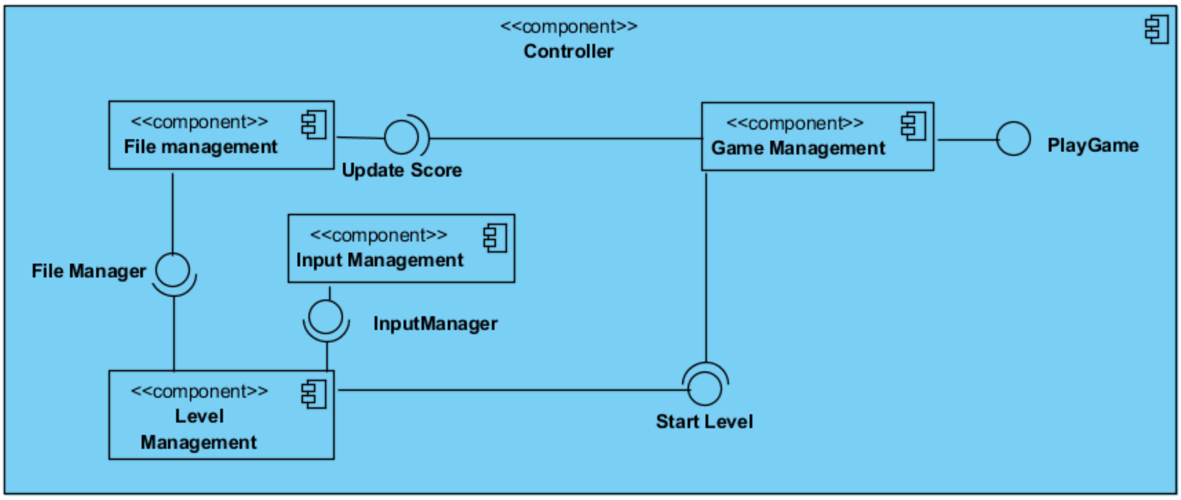
ModelView Component is providing interface for model views and it gets update according to changes in model subsystem from Controller subsystem

A new game can be instantiated only through the **Menu Component.** Menu Component can be invoked by the signals from Controller Subsystem. Those signals are “pause” and “exit game”. Menu Component has a functionality that starts a new game through Menu Class which invokes Game Management subsystem by calling its methods that starts a new game. Façade Design is applied to reduce coupling and increment coherence.

Frame Component provides user interfaces for How to Play and High score table. Frame component can only be invoked by Menu Component.

Settings Component provides settings for the user. The purpose of the settings component is keeping the user settings and changing the settings if the changes are applied. Settings component can only be invoked by Menu Component.

3.2. Controller Subsytem



Controller Subsystem is responsible for controlling the game. This subsystem consists 4 major components:

1. **File Management**
2. **Game Management**
3. **Level Management**
4. **Input Management**

File Management component is responsible for saving the high scores of the players. It is invoked by Level management subsystem at the beginning of the game to get level objects and at the end of the game by Game Management subsystem to update highest score.

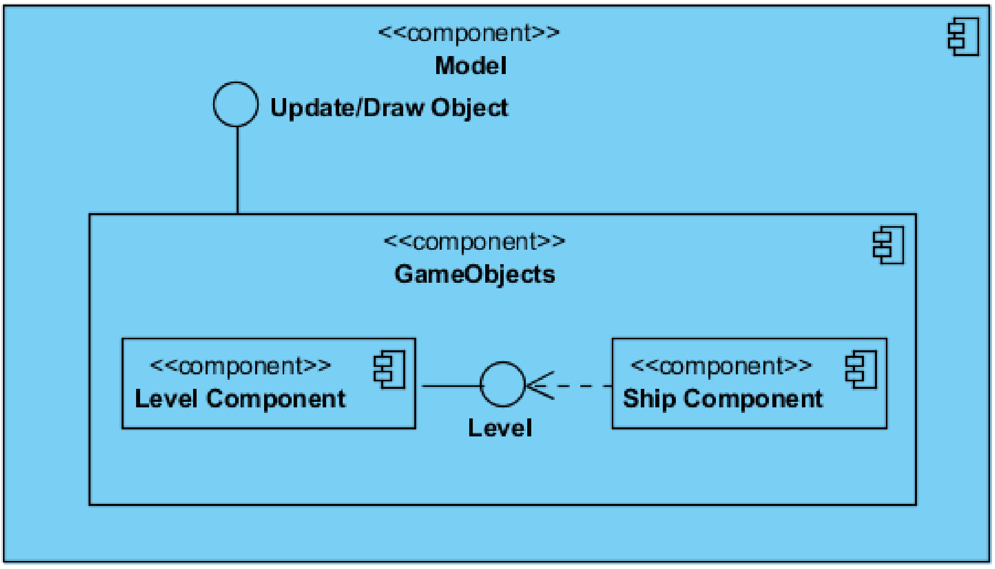
Game Management component's purpose is getting the play game input from the View subsystem and it starts a new game. Game management is only responsible for starting a new game, when the level is completed, Menu Component has been invoked during the game is paused or exited. Invoking

File Management when the new highest score has been achieved by updating the score.

Level Management component is responsible for controlling the model and tis components. It has **game loop,** level management components and models are updated constantly. Also, it updates the ModelView component of View subsystem. This component also checks the collisions because this component has collision checks. It frequently checks that whether there is a collision or not. Level management update the model objects according to Collision Manager.

Input Management handles the input from the user. When the user gives a specific input, level management has been notified and do a specific function accordingly to input.

3.3. Model Subsystem



Model Subsystem represents the model in the game. It consists of the Game Objects which includes several different types. Model has a main component which is called GameObjects and it represents the different type of game objects and entities. All game entities can only be updated and drawn by the method which is called from Controller Subsystem. Only Management component's level management component can modify the model component's instances. Model class has one component which consists of two subcomponents:

Game Entities: it represents the model objects

Level Component: Represents the bullets, shields and includes their operations.

Ship Components: Represents the player’s ship, enemy ships and the bonus ships.