

Gravity

Group 10

Daniel Walz

Jesper Ekhall

Lukas Kalinski

Fredrik Nordh

Alexander Nyberg

5.1 Class Responsibility Collaborator (CRC) Cards

5.1.1 View Module

<i>Distributor</i>	
Responsible for distributing the input game state to all suppliers implementing the SupplierInterface.	Game::State::GamePlay Game::State::Menu LocalRenderer
<i>LocalRenderer</i>	
Renders local graphics for one or at most two players.	Distributor Infrastructure::SDL

5.1.2 Game Module

<i>Game</i>	
Runs the main loop and forwards control to other game states.	Game::State::Menu Game::State::GamePlay

5.1.2.1 Engine Package

<i>Engine</i>	
Responsible for maintaining the game world, applying physical constraints to it and to trigger its visualization and event sounds.	Game::Engine::Physics::* Game::GameWorld::World Game::GameWorld::* Game::Player Registry::GameWorldMap::Registry

5.1.2.2 Engine Physics Package

<i>CollisionManager</i>	
Responsible for detecting and managing collisions between objects in the game world, provided a game world instance.	Game::Engine::Engine Game::Engine::Util::Sound Game::GameWorld::*
<i>GravityManager</i>	
Responsible for calculating and applying gravity affections for each gravity-affectable game world object.	Game::Engine::Engine Game::GameWorld::*

5.1.2.3 Engine Utilities Package

<i>Coord2d</i>	
Represents a coordinate in 2D space.	Game::GameWorld::*
<i>Sprite</i>	
A 2D image or animation that is used in the graphical presentation of the game. Will not contain the actual	Game::GameWorld::* Game::Menu::*

Sprite

image data but rather data about where the image(s) may be found.	
---	--

Sound

Abstract representation of a sound. Will not contain the actual sound data but rather data about where the sound may be found.	Game::Engine::Physics::CollisionManager
--	---

5.1.2.4 State Package***AbstractState***

Contains common functionality for a game state.	GamePlay Menu
---	------------------

GamePlay

The game state to be in when a game session is active.	Game::Engine::Engine View::Distributor Audio::Distributor
--	---

Menu

The state to be in when navigating through menus.	Game::Menu::* View::Distributor
---	------------------------------------

5.1.2.5 State Action Package***StartSinglePlayerGameAction***

Start a single player game on a selected map.	
---	--

PauseGameAction

Pause a game session.	
-----------------------	--

StartMultiPlayerGameAction

Start a multiplayer game on a selected map.	
---	--

ShowHighScoreAction

Show high score screen/menu.	
------------------------------	--

AbstractShipAction

Common properties and operations for ship actions.	
--	--

ShipLeftAction

Turn a ship left.	
-------------------	--

ShipRightAction

Turn a ship right.

ShipThrottleAction

Throttle a ship.

ShipWeaponAction

Fire a ship's weapon.

ShowGameControlsMenuAction

Load the game controls configuration menu.

QuitGameAction

Quit the whole game application.

UnpauseGameAction

Unpause a game session.

EndGameSessionAction

End a game session.

ShowMainMenuAction

Show the main menu.

SaveGameControlsAction

Save game controls when set.

ShowMapChoiceMenuAction

Show map choice menu.

5.1.2.6 Menu Package***AbstractMenu***

Contains common properties and operations for a menu.

Item

MainMenu

The main menu structure.

PauseMenu

The pause menu structure.

HelpMenu

The help menu structure.

HighScoreMenu

The high score menu structure.

Registry::HighScore::Registry

MapChoiceMenu

The map choice menu structure.

Registry::GameWorldMap::Registry

GameControlsMenu

The game controls configuration menu structure.

Registry::Config::Registry

5.1.2.7 GameWorld Package

World

Representation of the whole game world, containing all world objects that are supposed to exist at a certain moment during a game session.

AbstractWorldObject

AbstractWorldObject

Contains common behavior and properties of game world objects.

Game::Engine::Util::Sprite
Game::Engine::Util::Coord2d

Ship

Representation of the ship that a game player will control.

AbstractProjectile

Contains common behavior and properties of weapon projectiles fired by a ship.

Game::Engine::Util::Sound

LaserProjectile

Representation of a laser projectile, which is not affected by gravities.

MissileProjectile

Representation of a gravity-affectable missile projectile.

Asteroid

Representation of an asteroid flying around randomly in the game world.

<i>Planet</i>	
Representation of a planet.	

<i>AbstractItem</i>	
Contains common behavior and properties of items occurring in the game world.	

<i>FuelItem</i>	
Representation of an item that contains a fuel powerup.	

<i>WeaponItem</i>	
Representation of an item that contains a ship weapon upgrade.	

5.1.2.8 Player Package

<i>AbstractPlayer</i>	
Holds common properties for an abstract player.	Game::GameWorld::Ship

<i>SinglePlayer</i>	
Holds statistics about a player when playing in single player mode.	

<i>MultiPlayer</i>	
Holds statistics about a player when playing in multiplayer mode.	

5.1.3 Registry Module

<i>AbstractRegistry</i>	
Common functionality for the registry pattern.	

5.1.3.1 Config Package

<i>Registry</i>	
Registry of the game configuration settings.	Persistence::File::FileManager

5.1.3.2 HighScore Package

<i>Registry</i>	
Manages player high scores.	Persistence::File::FileManager

5.1.3.3 GameWorldMap Package

<i>Registry</i>	
Provides an interface for accessing game world map	Persistence::File::FileManager

<i>Registry</i>	
data.	

5.1.4 Persistence Module

5.1.4.1 File Package

<i>FileManager</i>	
Provides functionality to create, read, modify and write files in the file system.	

<i>AbstractFile</i>	
Contains common operations and functionality of a file system file.	

<i>BitmapFile</i>	
Represents a bitmap image file and provides methods to read such files.	

<i>TextFile</i>	
Represents a plain text file and provides methods to read and write such files.	

<i>Mp3File</i>	
Represents a mp3 sound file.	

5.1.5 Audio Module

<i>Distributor</i>	
Responsible for distributing the input sound play orders to all suppliers implementing the <code>SupplierInterface</code> .	<code>LocalPlayback</code>
<i>LocalPlayback</i>	
Plays sounds locally on the computer that is running the game engine.	<code>Infrastructure::FMOD</code>

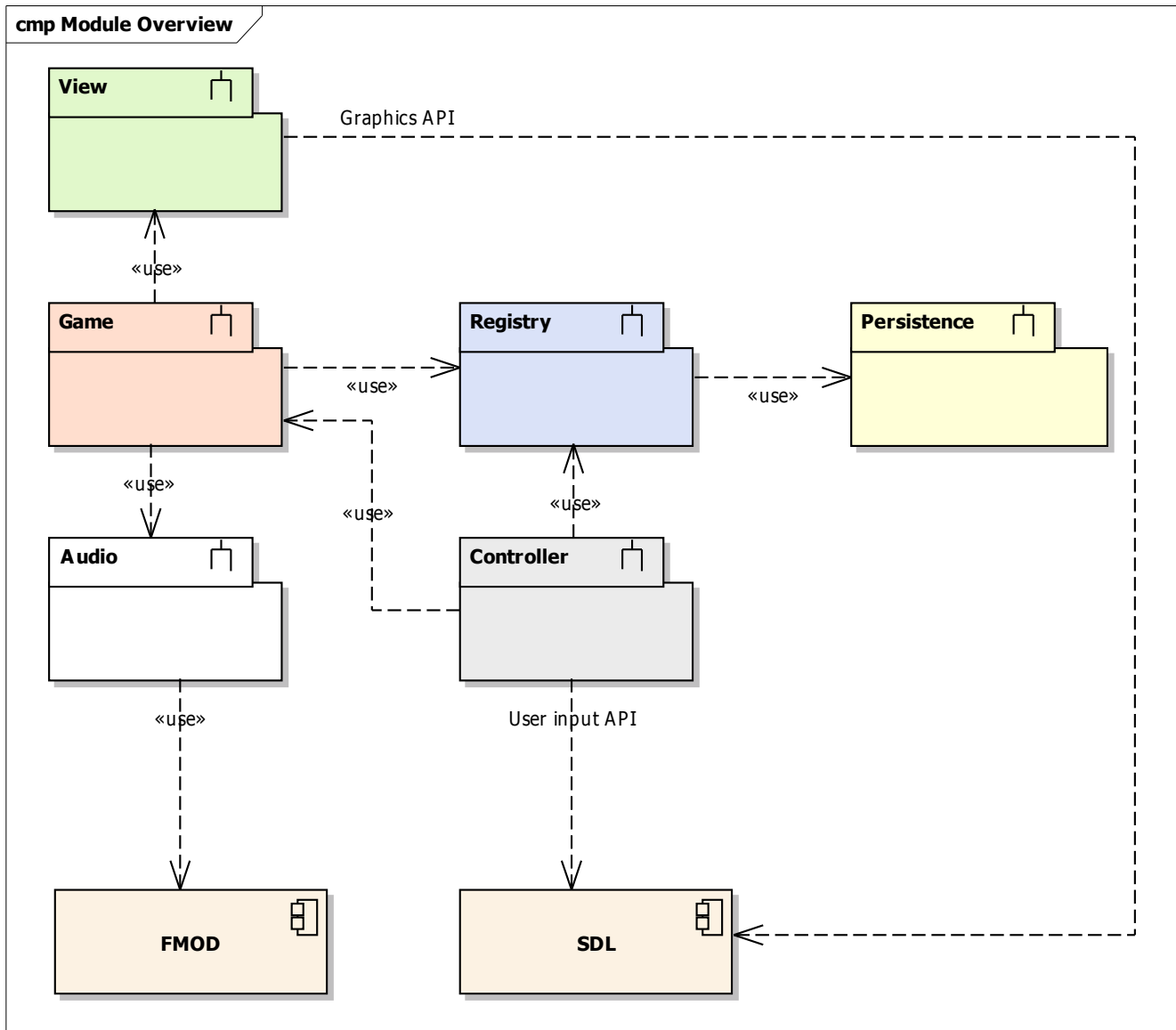
5.1.6 Controller Module

<i>InputManager</i>	
Monitors the statuses of all available input listeners and triggers corresponding game actions, according to the translation map(s) found in the <code>Registry::Config</code> package.	<code>KeyboardListener</code> <code>Game::State::Action::*</code>
<i>KeyboardListener</i>	
Takes keyboard input from SDL.	<code>Infrastructure::SDL</code> <code>Registry::Config::Registry</code>

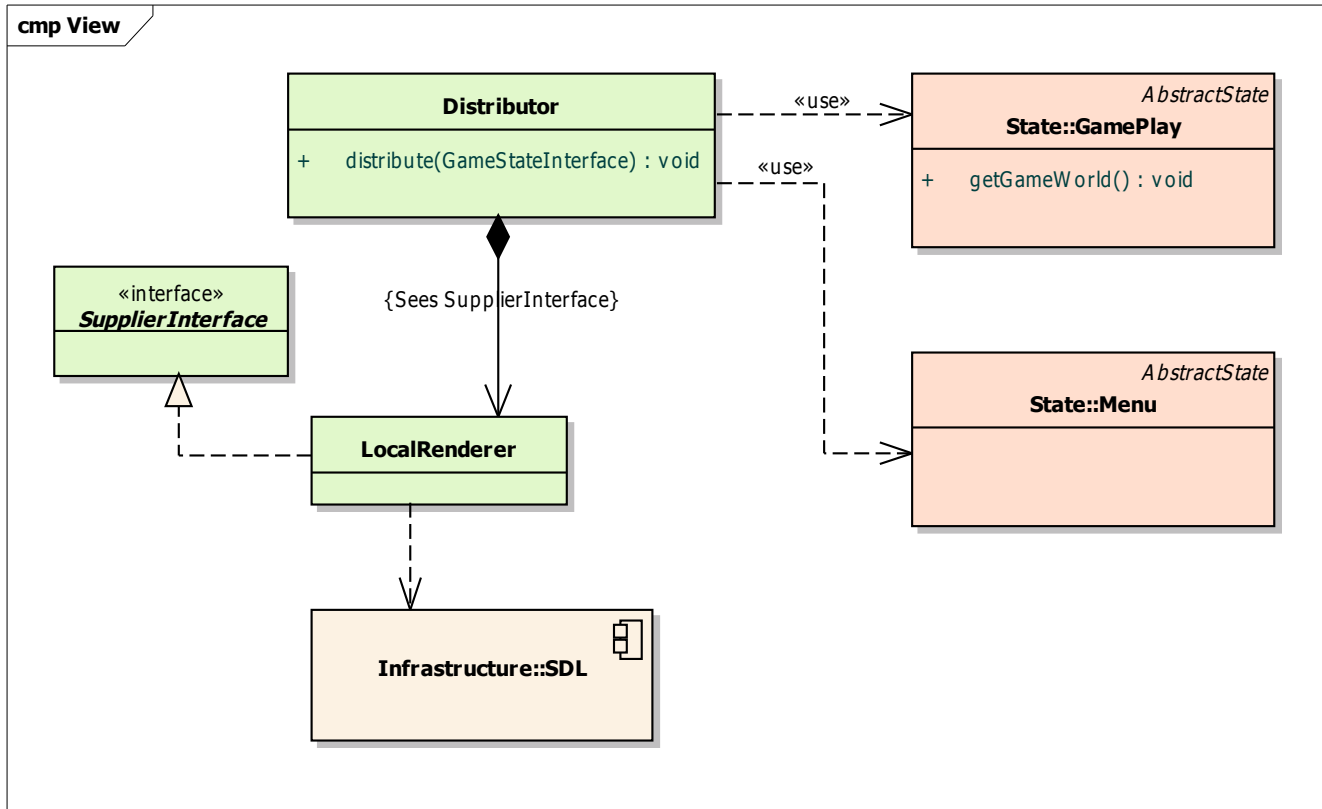
5.2 Class Diagram

First in this section there is a module overview which shows the modules we shall have. Then there are subsections that show all modules and what they shall contain. Everything is presented in Unified Modeling Language (UML). For more details about the classes see section 5.5.

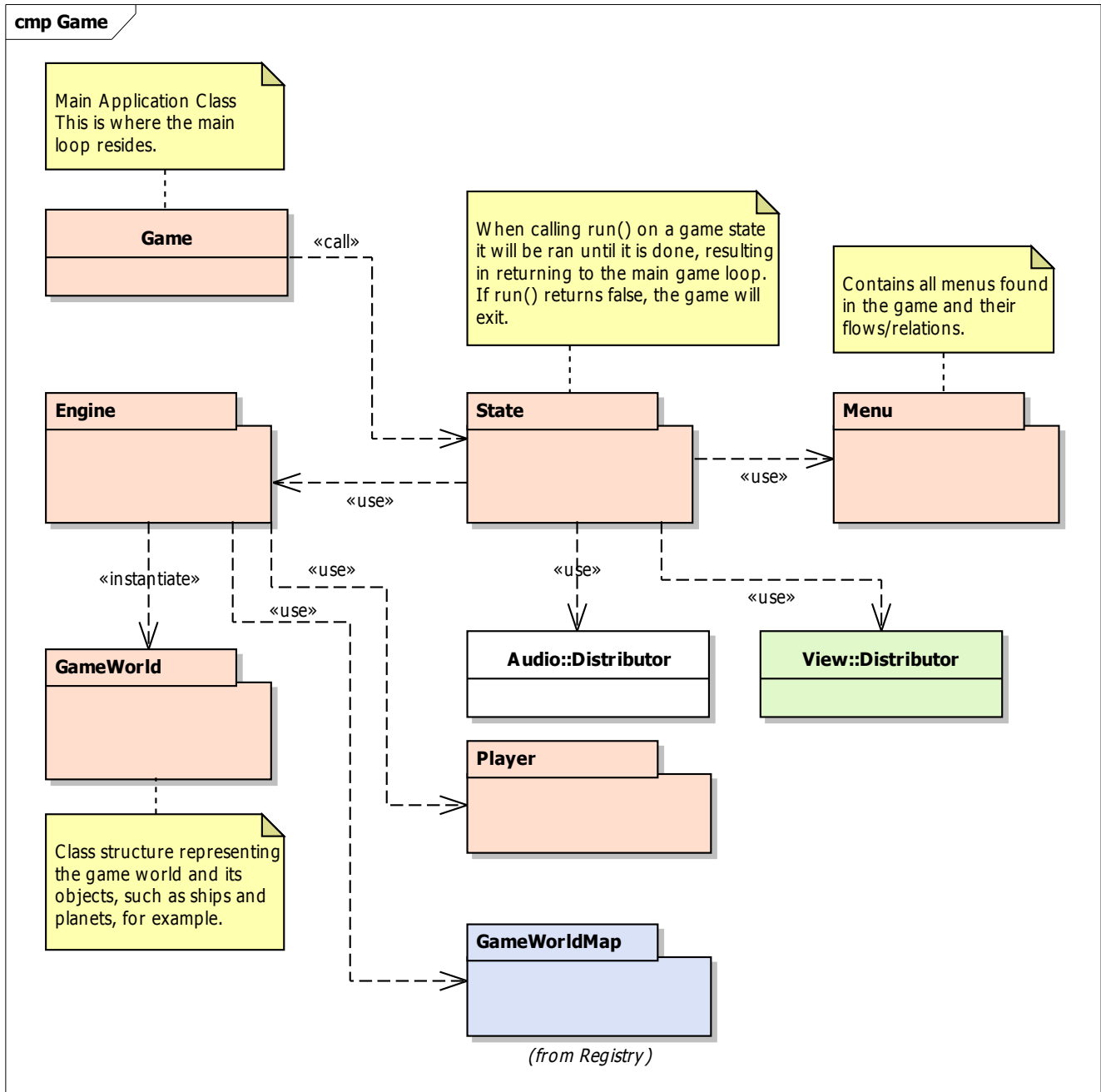
5.2.1 Module Overview



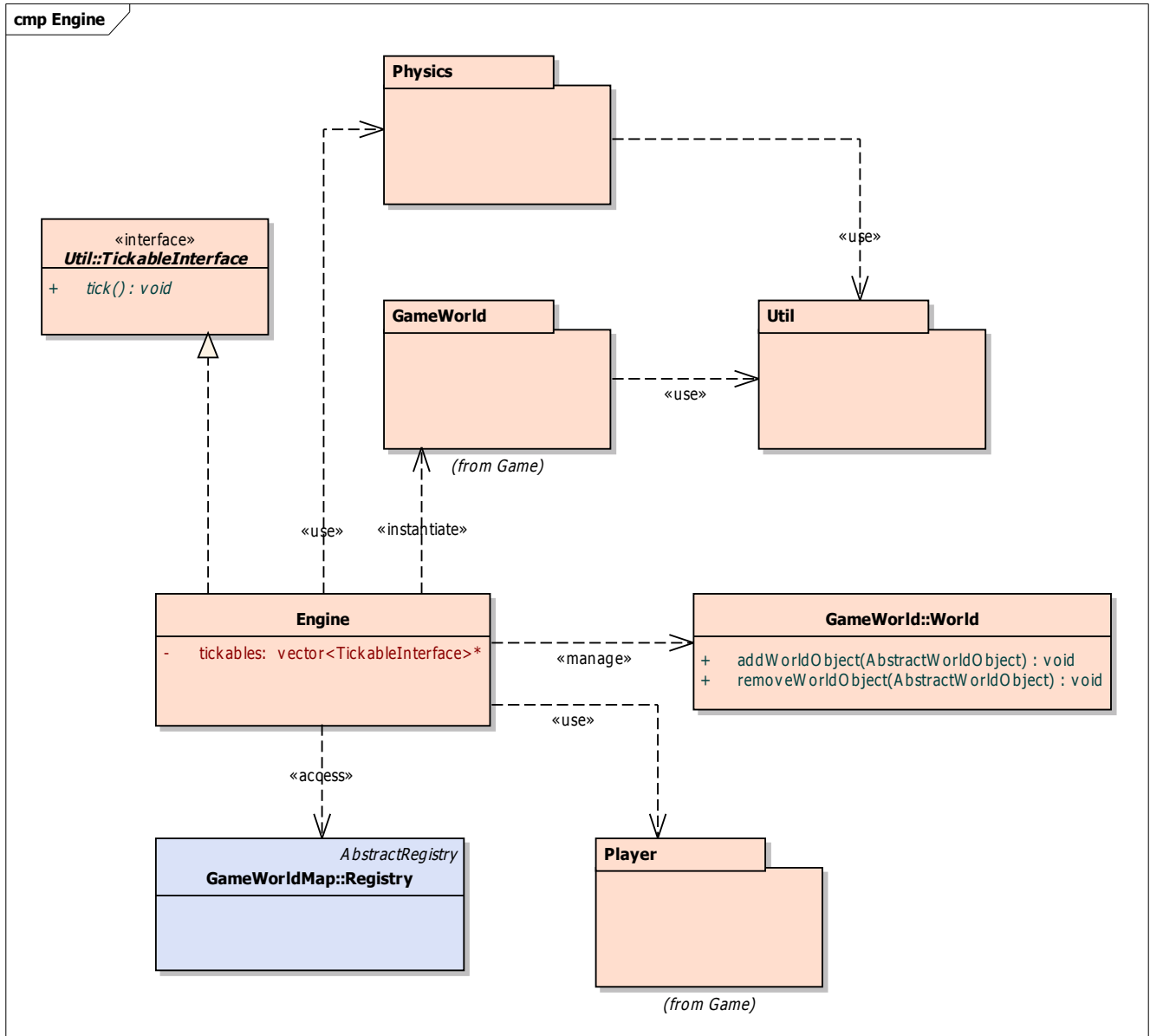
5.2.2 View Module



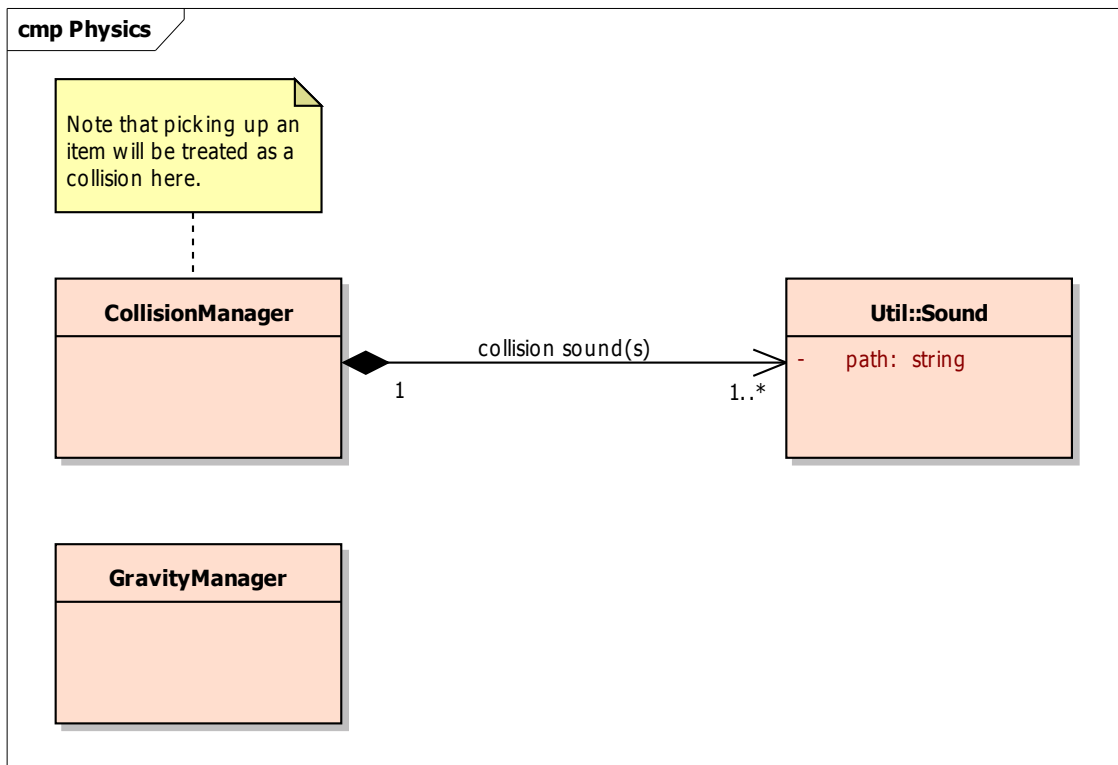
5.2.3 Game Module



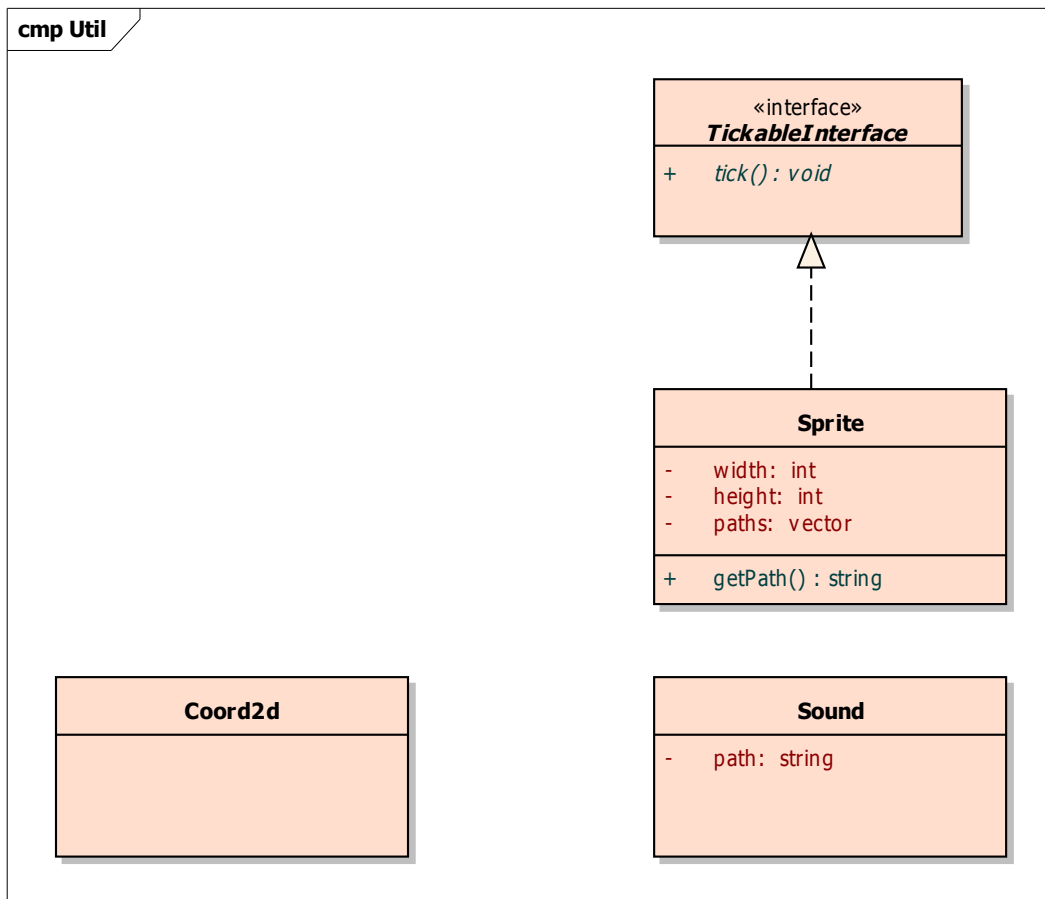
5.2.4 Game Module: Engine Package



5.2.5 Game Module: Engine Physics Package

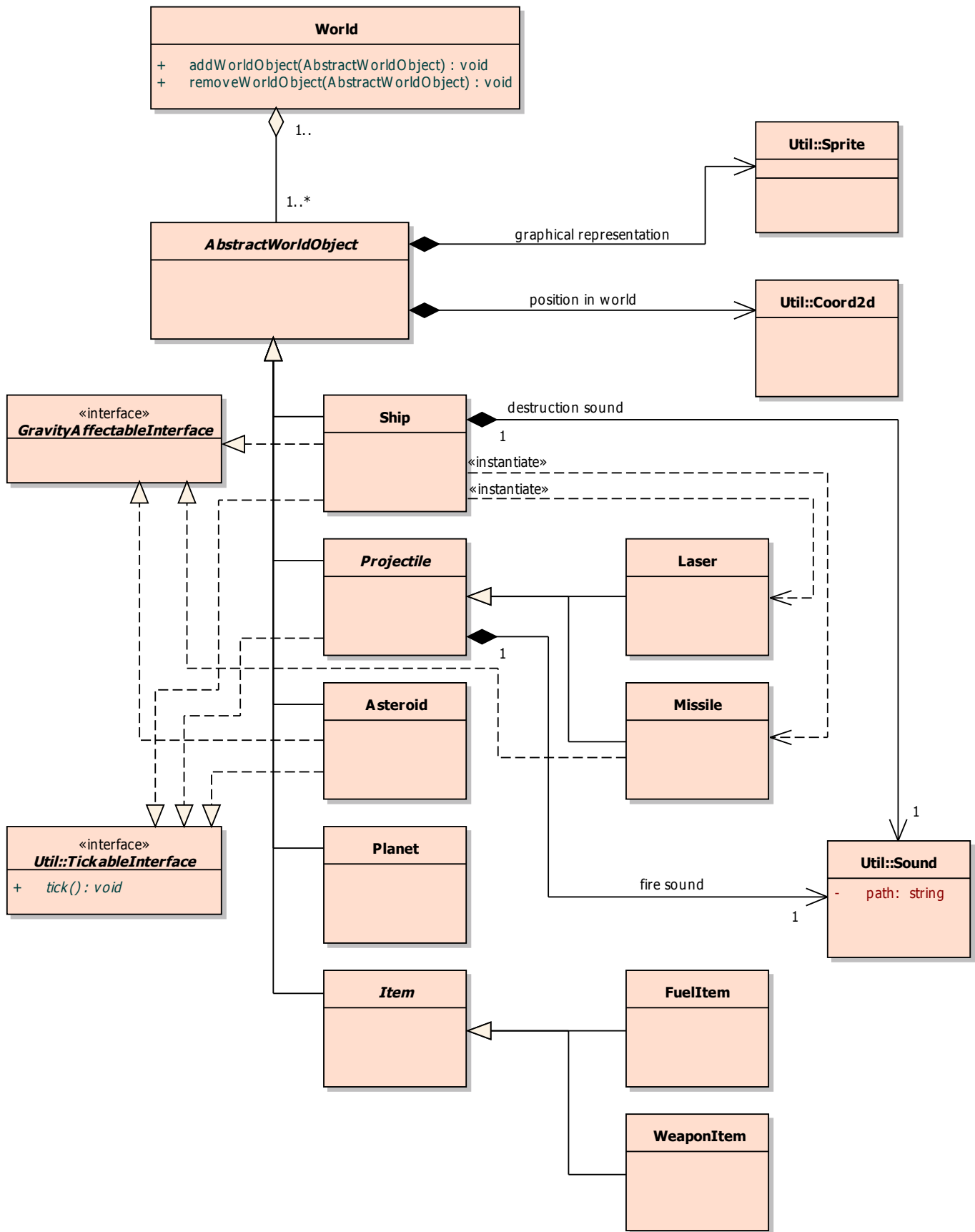


5.2.6 Game Module: Engine Utilities Package

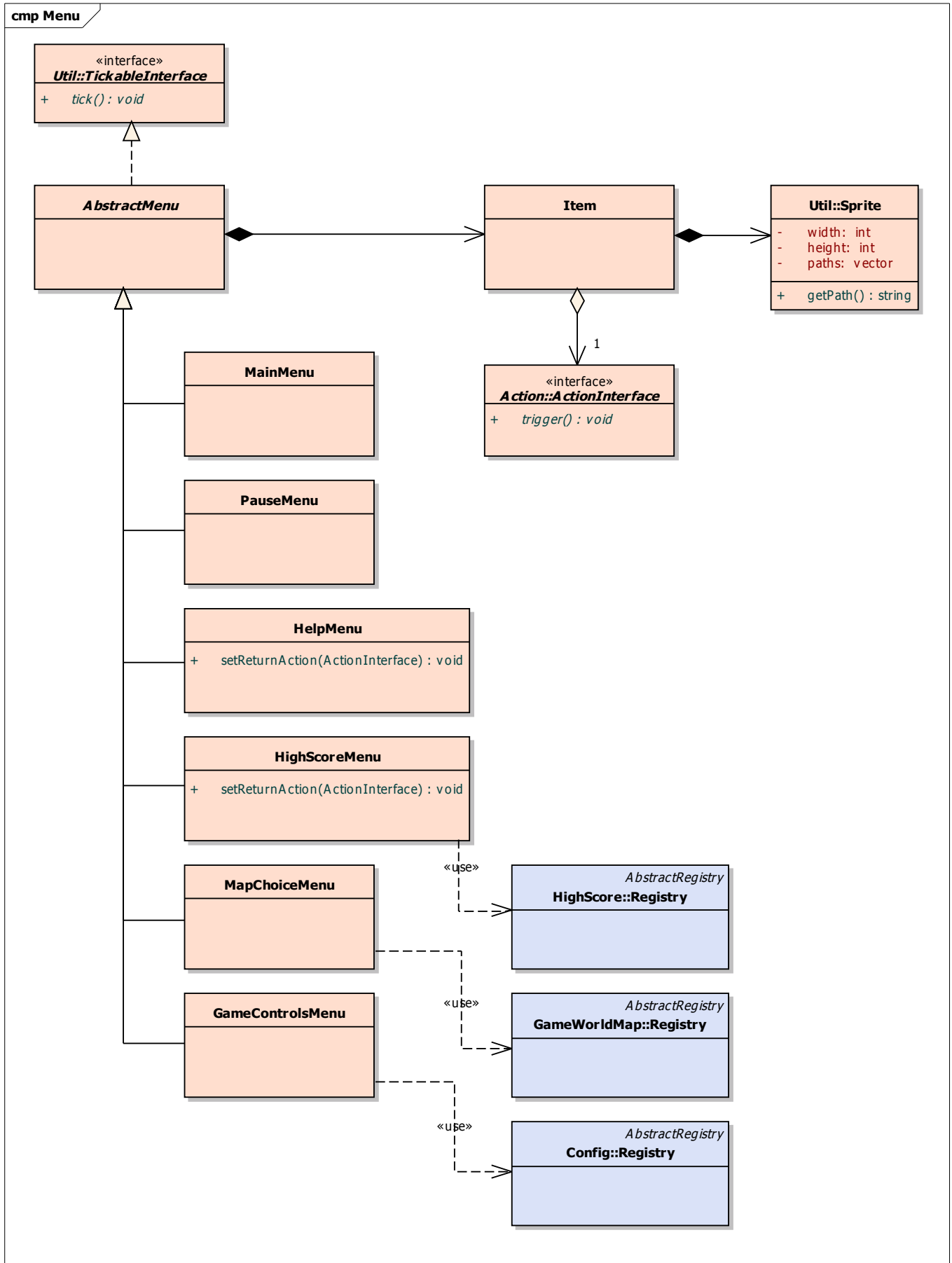


5.2.7 Game Module: Game World Package

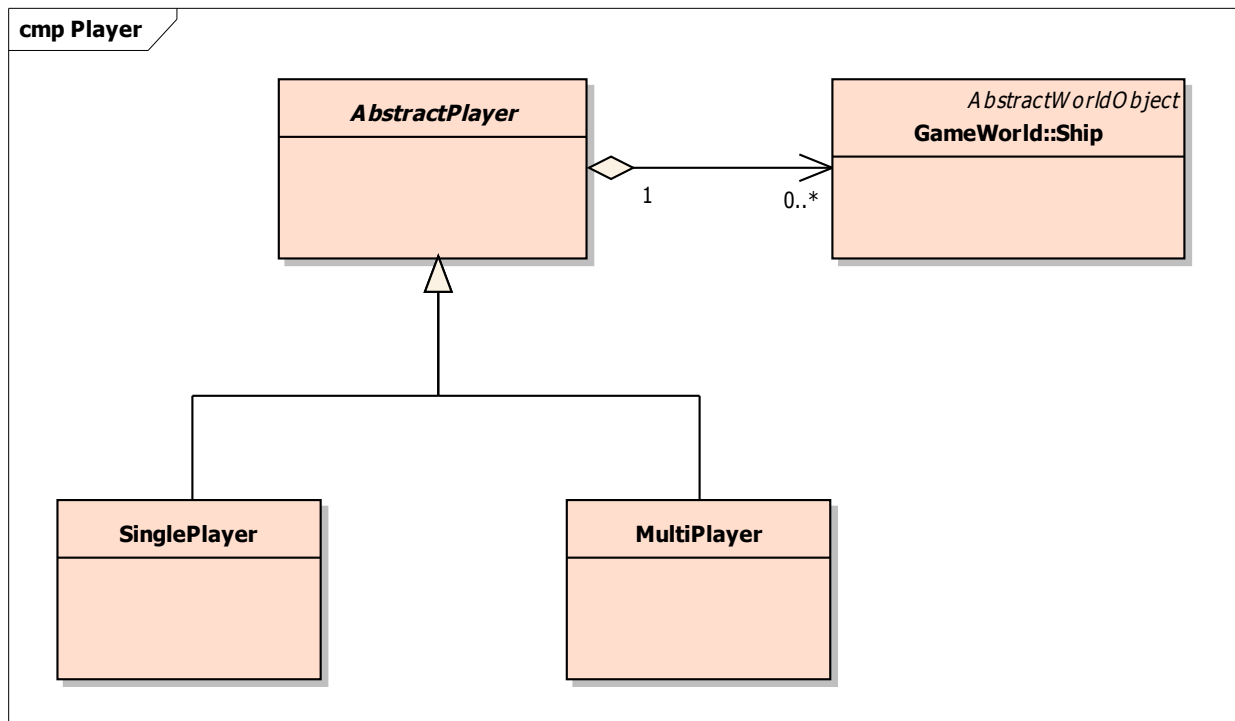
cmp GameWorld



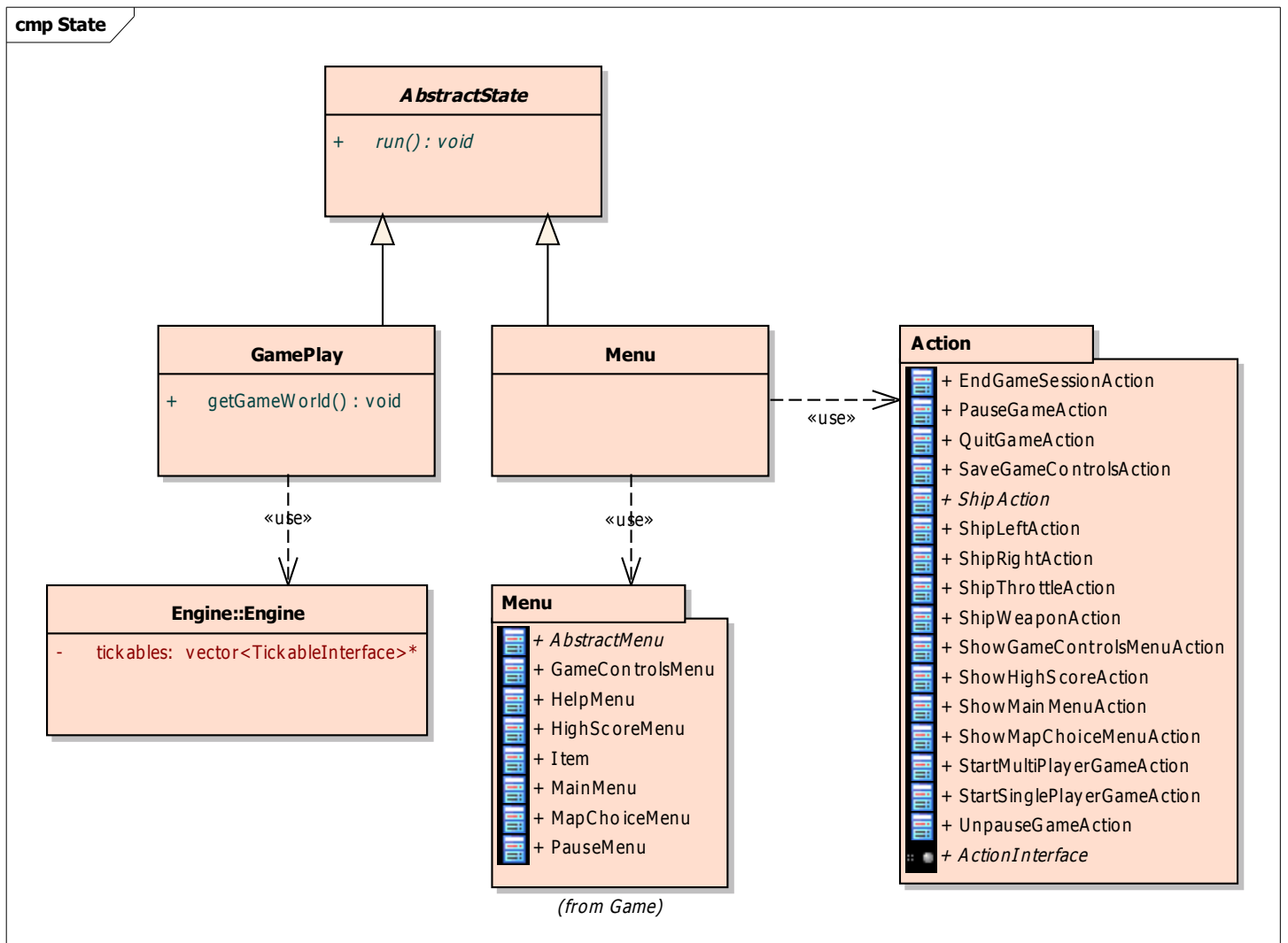
5.2.8 Game Module: Menu Package



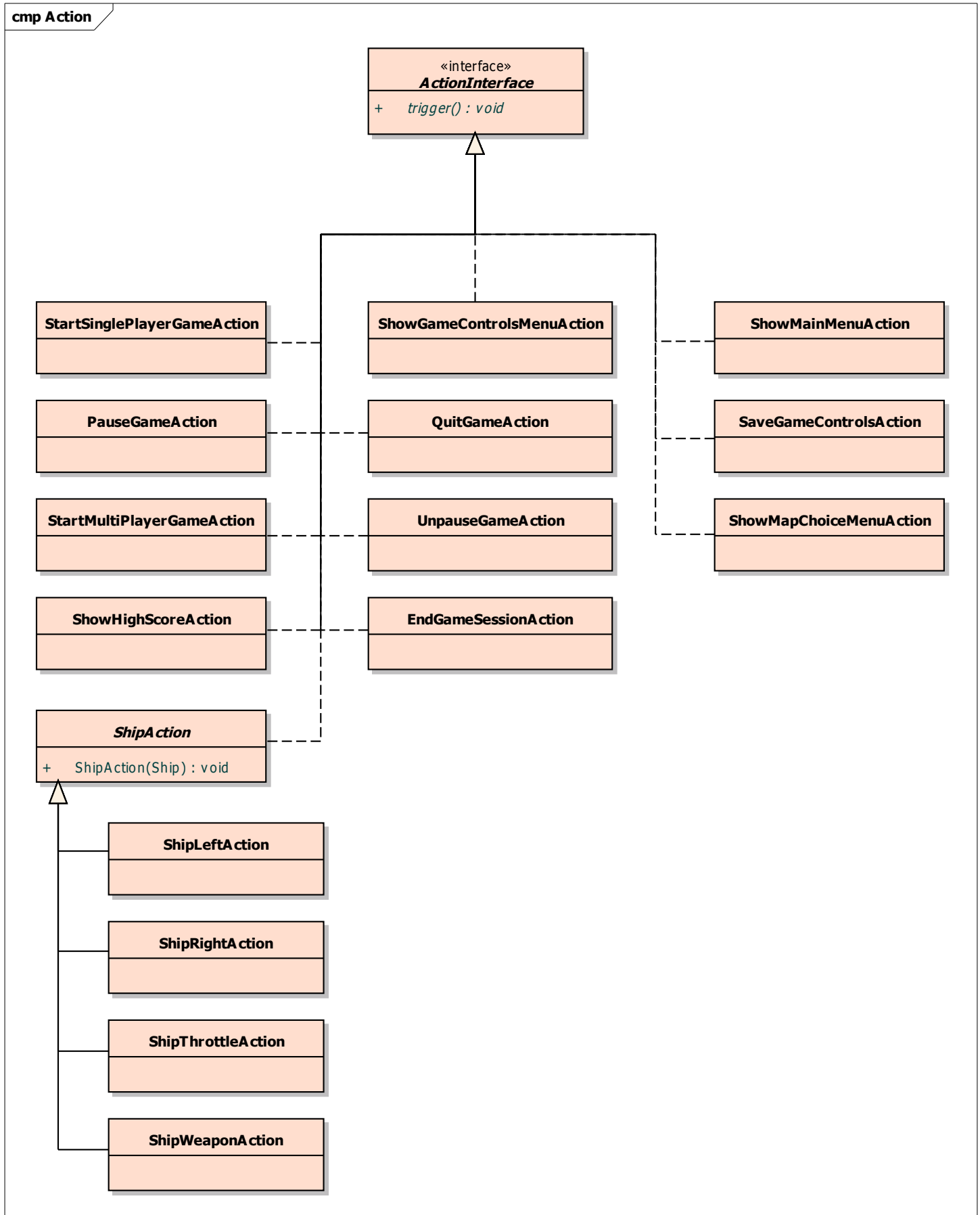
5.2.9 Game Module: Player Package



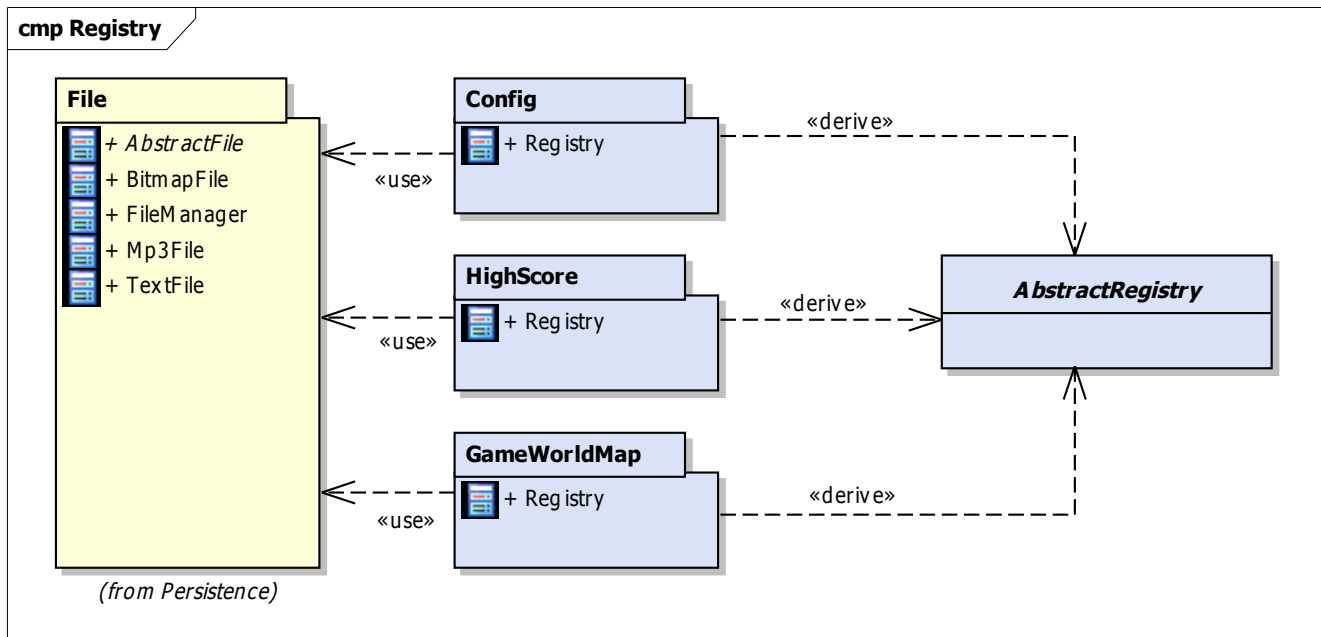
5.2.10 Game Module: State Package



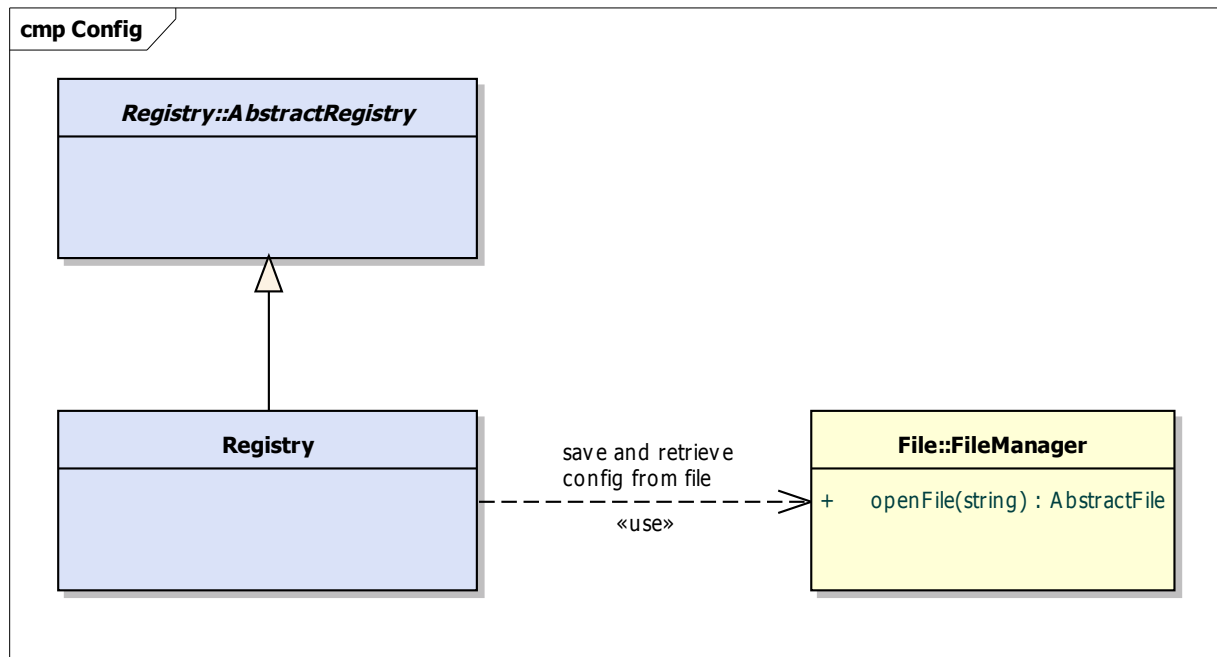
5.2.11 Game Module: State Action Package



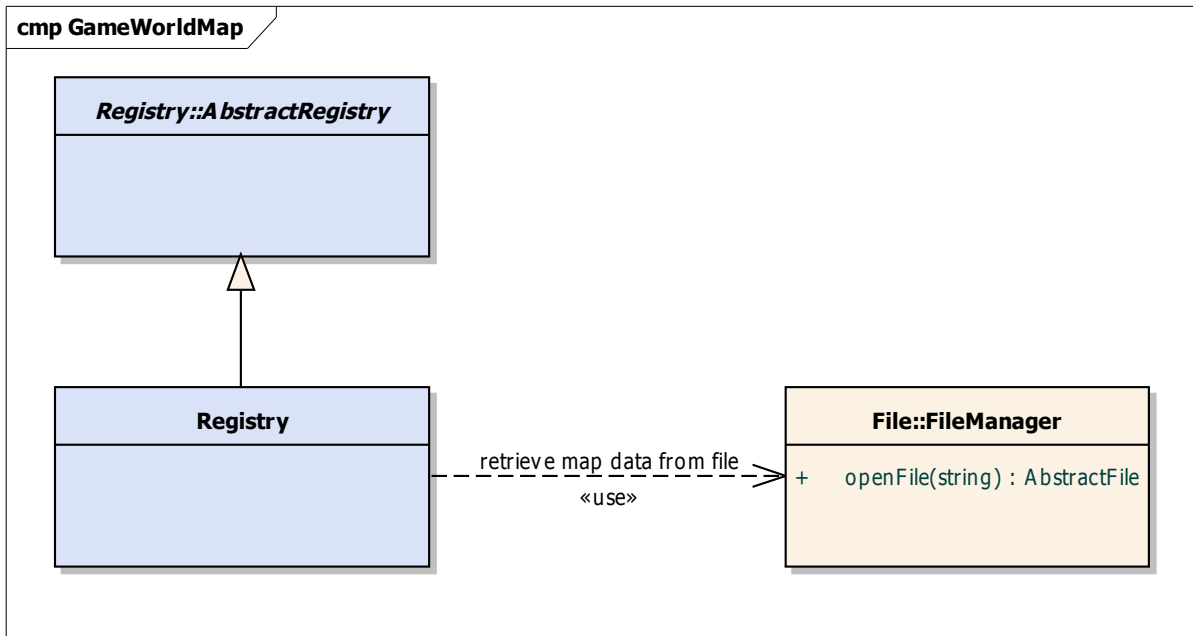
5.2.12 Registry Module



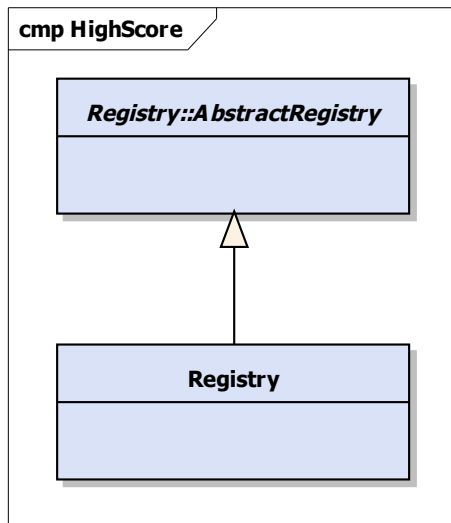
5.2.13 Registry Module: Game Configuration Package



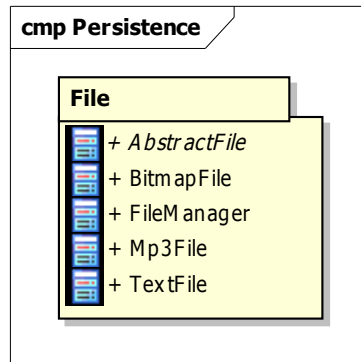
5.2.14 Registry Module: Game World Map Package



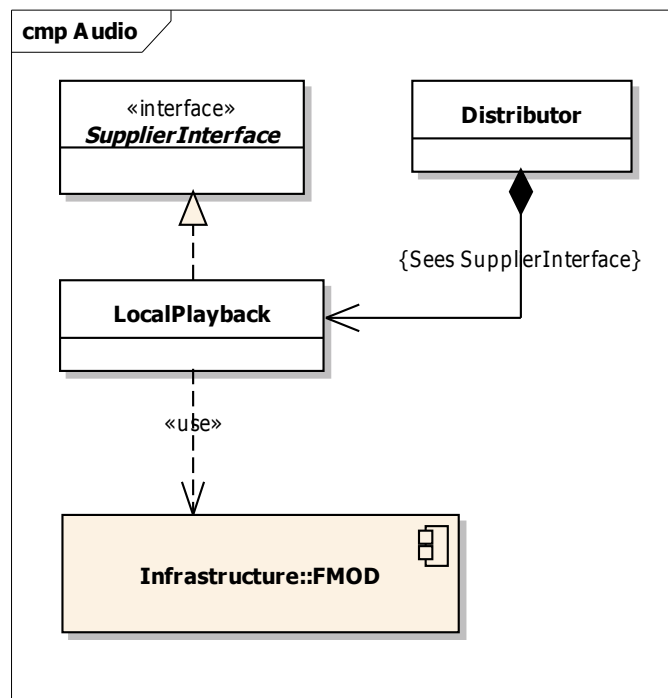
5.2.15 Registry Module: High Score Package



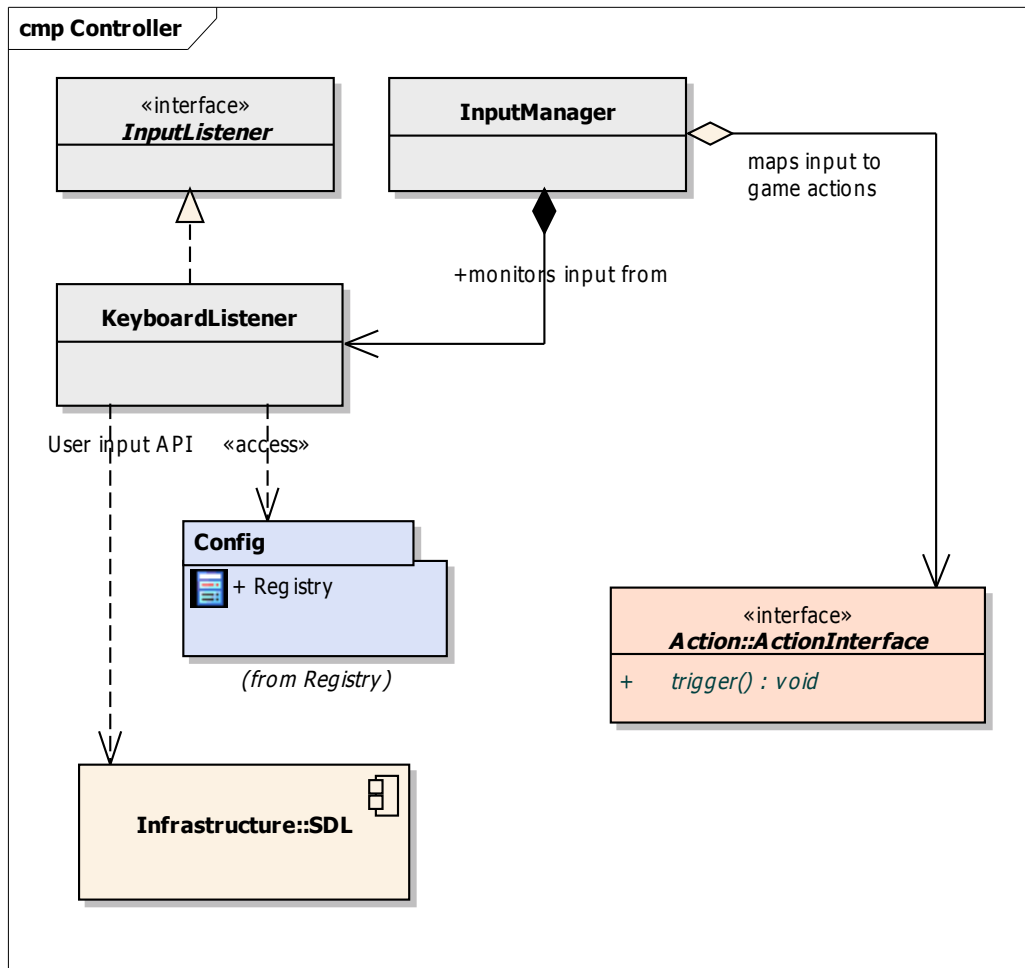
5.2.16 Persistence Module



5.2.17 Audio Module



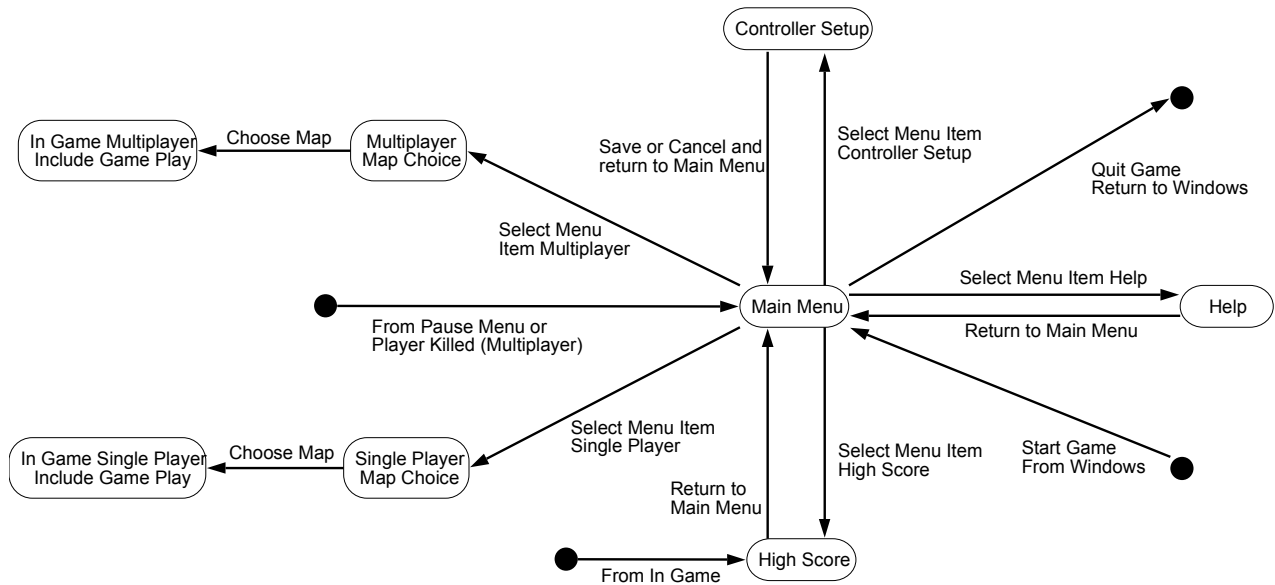
5.2.18 Controller Module



5.3 State Charts

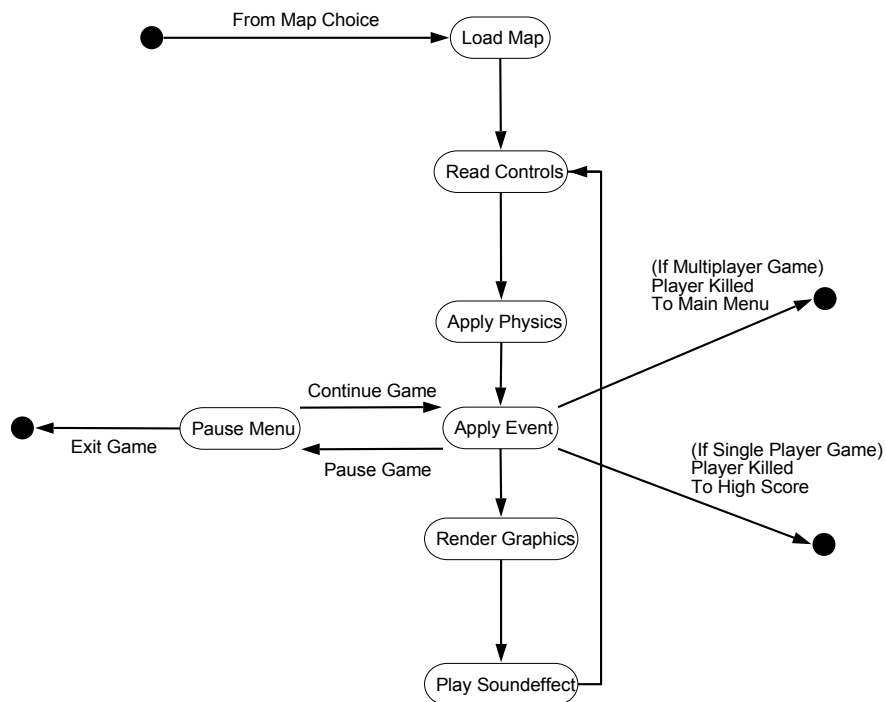
We have two main flow of control in this game. The first part is before starting the actual game and navigating the menus. The second part is for what happens while actually playing the game.

5.3.1 Pre-Game Flow



From these menus it's possible to see your high score, get help and setup keyboard for how to control the game. In the menus it's possible to say how you want to play the game, such as what map to play at and also if you want to play multiplayer or single player game.

5.3.2 In-Game Flow

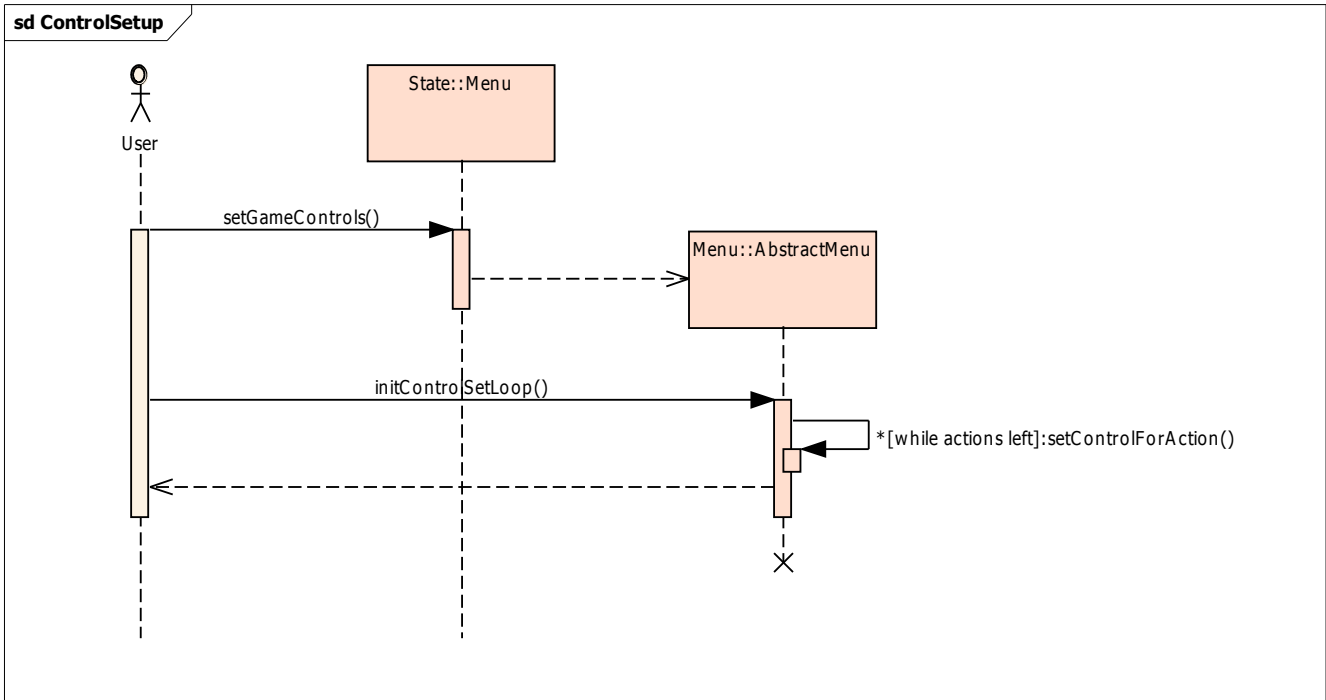


While playing the game there is a main loop that takes care of reading the keyboard, calculating game physics, playing sound and applying game logics.

5.4 Interaction Diagrams

There are two interaction diagrams which show the control setup and the game start sequence.

5.4.1 Control Setup



5.4.2 Game Start

