System Design Document
SDD

Project Cosmic Shooter

Team 30
Anton Andersson, Jonatan Arvidsson, Adrian Danielsson,
Linn Lindgren och Simon Bergström

Contents

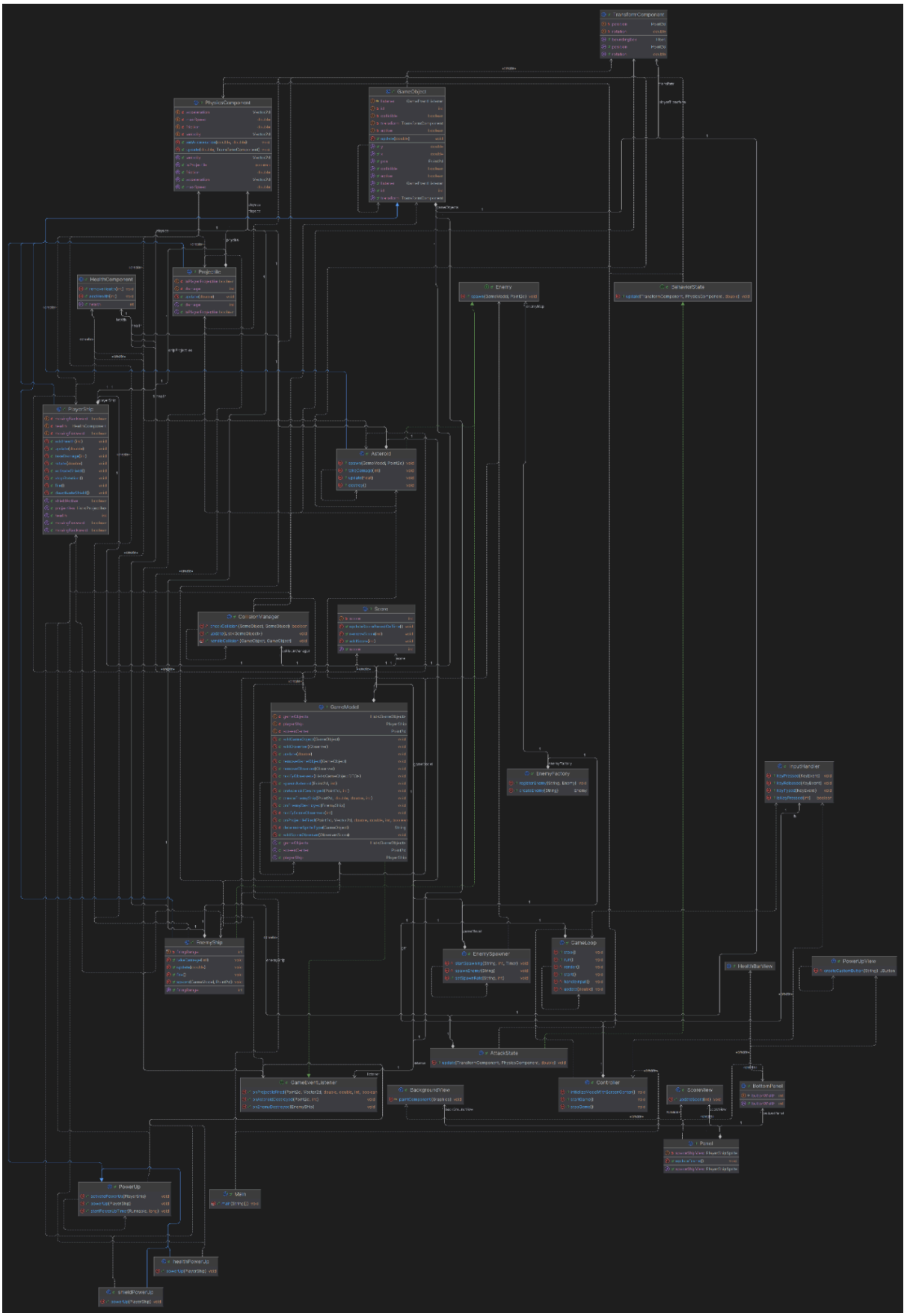
1. Introduction

1.1 Purpose and Scope

The developed system for the retro inspired game Cosmic Shooter, is designed to be as modular and object oriented as possible. A system designed by these mentioned principles, facilitates an easier implementation, modification or removal of new modules in future extensions.

An implementation of the architectural pattern, model-view-controller (MVC), was also appropriate for the project system, because the system is best designed with clear and separate modules for the data and user presentation. This improves the modularity of the application.

2. Overview and UML-diagram



2.1 Model

The model consists of several classes which are responsible for managing the logic of the application. The GameModel class manages the overall logic and state of the game. The GameObject class is an abstract class that different objects in the game inherit, e.i. Asteroid, EnemyShip, Playership and PowerUp. This makes the game more expandable if more game objects are implemented. The PowerUp class itself is abstract and inherited by other power up classes, i.e. ShieldPowerUp and HealthPowerUp. The model also consists of many other game components that contain different game logic.

2.2 View

The view module uses java swing to display the different game components which are divided into separate classes and updated/rendered based on the state of our model. It also contains a few panels and a background which aren't dependent on the model.

2.3 Controller

The controller mainly handles the keyboard input for the movement of the player and activation of powerups. It also consists of a game loop class which contains the logic for actively updating the game model and rendering the view multiple times a second.

3. Detailed Design

3.1 Factory

3.2 Observer