

THE TRAFFIC WARRIOR!

Amir Abas Kabiri Zamani
Mohamad Reza Emami Naseri

OVERVIEW

- ◉ How to use Design Patterns?
- ◉ Project Description
- ◉ Functional Requirements
- ◉ Non-Functional Requirements
- ◉ Use Case Diagram
- ◉ Class Diagram
- ◉ Used Design Patterns

HOW TO USE DESIGN PATTERNS?

Pattern Smell

- **Assessing Requirements:**
 - Non-Functional Requirements:
 - Portability
 - Theme Changing
 - Etc.
 - Functional Requirements:
 - Firing!
 - Moving
 - Etc.

HOW TO USE DESIGN PATTERNS?

Classification

- **Detecting The Classification:**
 - Creational
 - Structural
 - Behavioral

HOW TO USE DESIGN PATTERNS?

Right Pattern

- **Detecting The Right Design Pattern:**
 - Intent
 - Applicability
 - Liabilities

PROJECT DESCRIPTION

- ◉ Single-Client game
- ◉ The client might take care of the machine versus obstacles
- ◉ Player's authorities:
 - Shooting
 - Moving

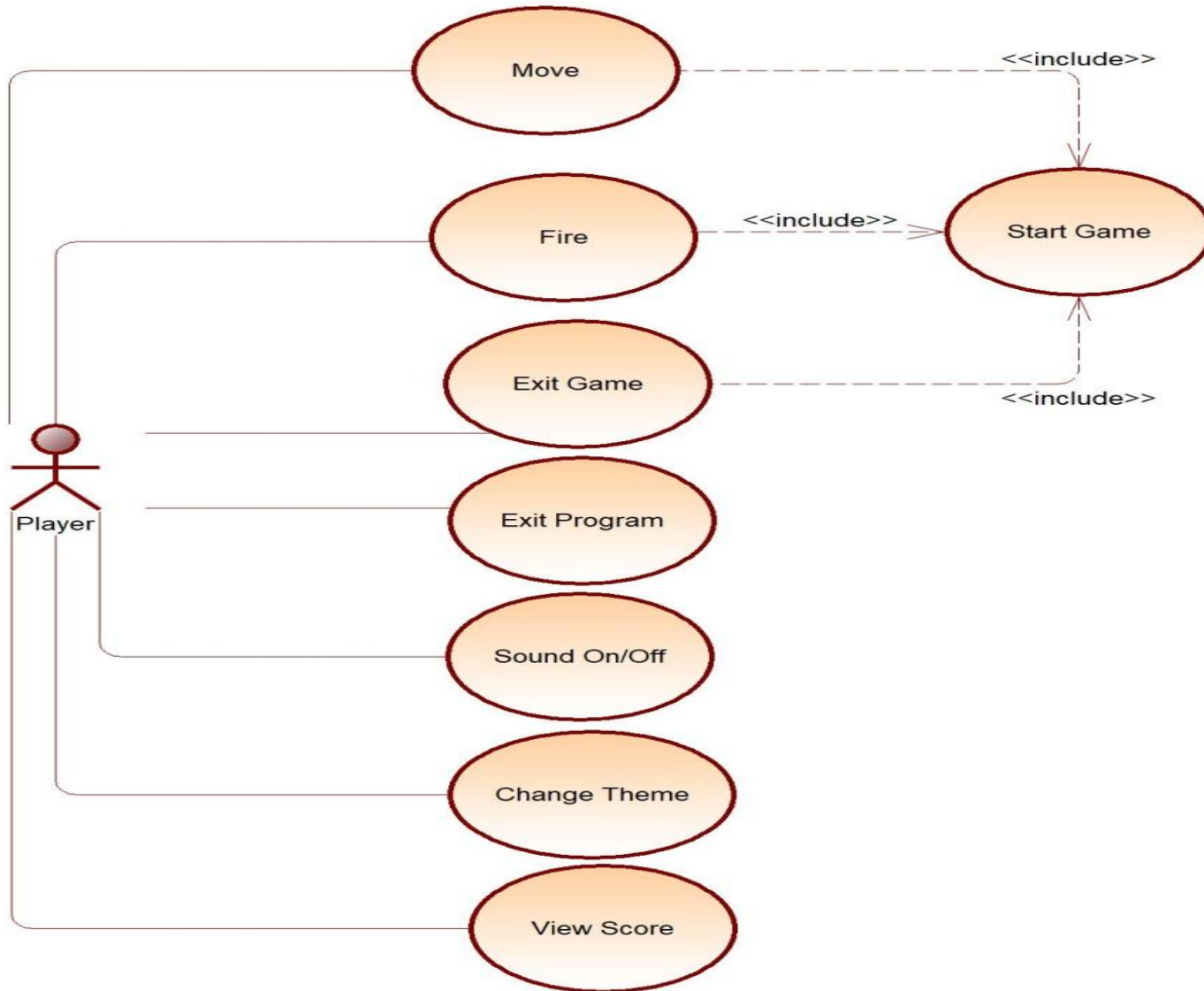
FUNCTIONAL REQUIREMENTS

- ◉ Shooting different types of bullet
- ◉ Avoiding accident by moving the machine
- ◉ Creating random obstacles
- ◉ Creating random Upgrade Items

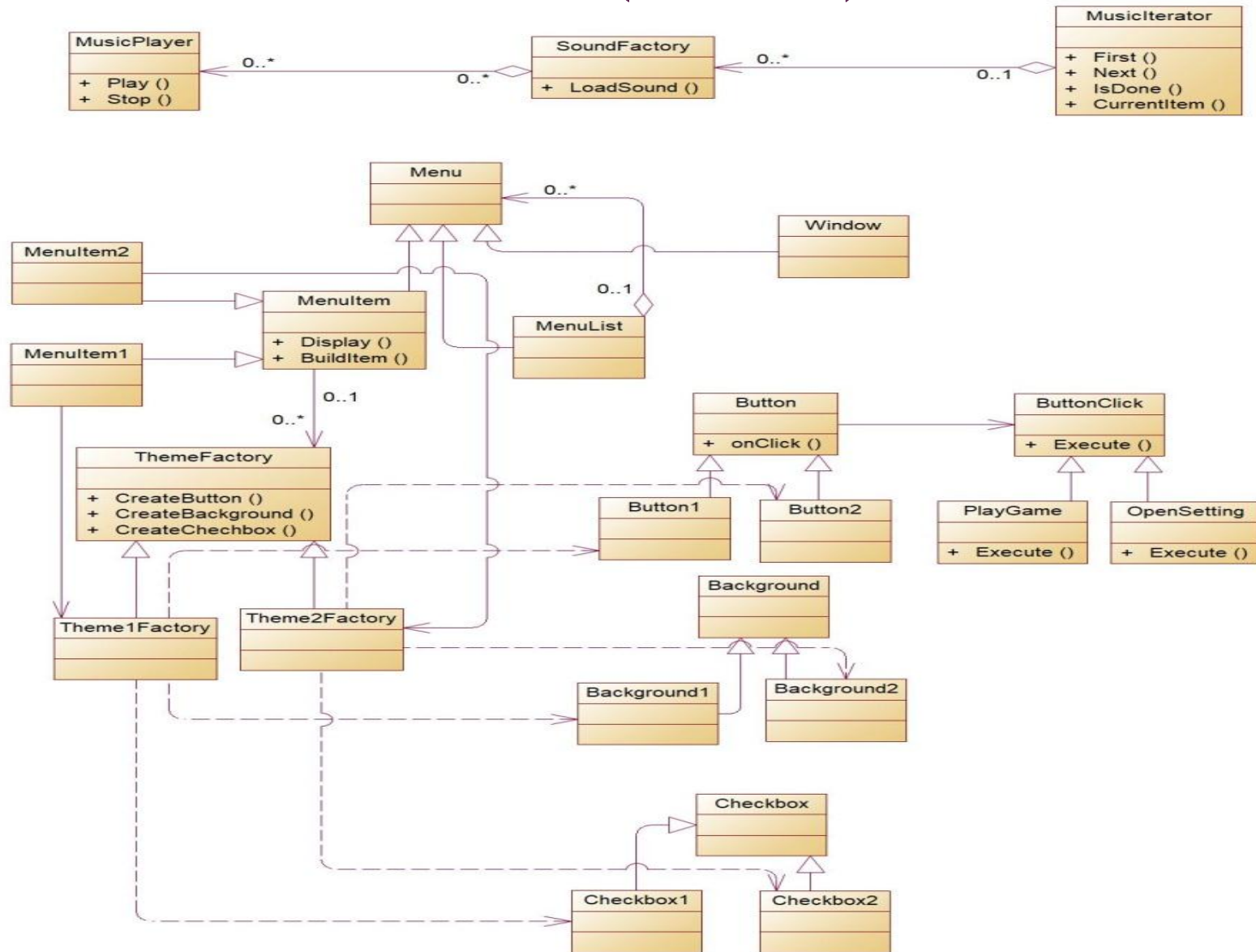
NON-FUNCTIONAL REQUIREMENTS

- ◉ Capability of changing menu's theme:
 - Abstract Factory Pattern
- ◉ Capability of shooting different Bullets and gaining different Armors:
 - Strategy Pattern
- ◉ Capability of streaming different sounds simultaneously:
 - Flyweight Pattern
- ◉ Etc.

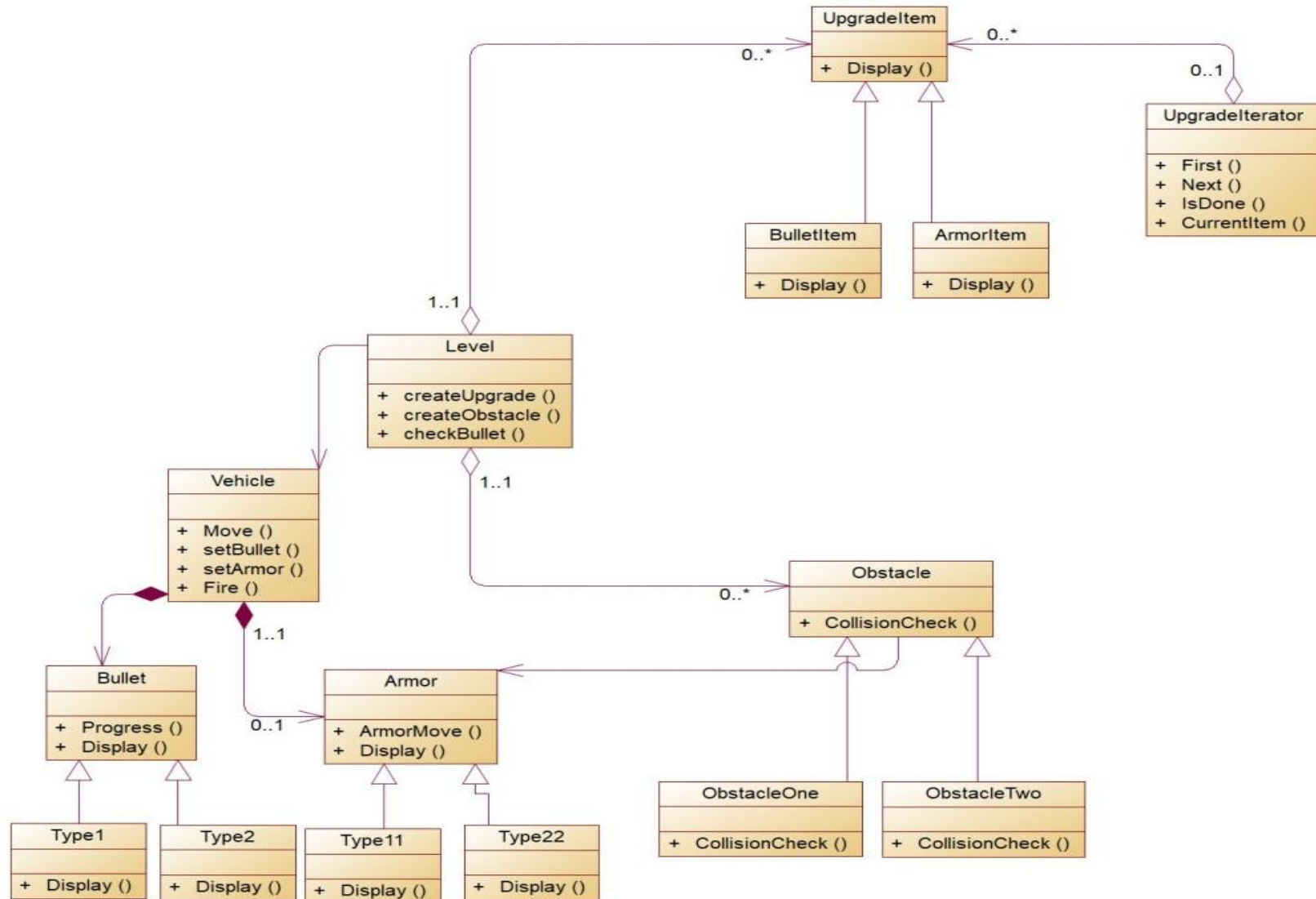
USE CASE DIAGRAM



CLASS DIAGRAM (MENU)



CLASS DIAGRAM (IN-GAME)



USED DESIGN PATTERNS - STRATEGY

◉ There exist two behavior relating to the machine within game:

- 1. Bullet
- 2. Armor

Each contains two different types.

USED DESIGN PATTERNS - FLYWEIGHT

- ◉ We might use Flyweight Pattern for the sounds playing within the game:
 - 1. Shooting sound
 - 2. Collision sound
 - 3. Music sound
 - 4. Etc.

USED DESIGN PATTERNS - ITERATOR

- ◉ We might use an iterator for the music player, containing some sort of options:
 - 1.First
 - 2. Next
 - 3. IsDone
 - 4. Etc.

USED DESIGN PATTERNS - ABSTRACT FACTORY

- ◉ We might use an abstract factory method while we intend to create a set of MenuItem's simultaneously and from corresponding style.

USED DESIGN PATTERNS - COMPOSITE

- ◉ We might use the composite pattern for the nested windows of our Menu. Since there may exist several nested windows, clicking a button, directing the client to a new window which has the same properties as the former window.

USED DESIGN PATTERNS - COMMAND

- ◉ We might use the command pattern to delegate the operation of clicking a Button and entering new window, to another class named ButtonClick.

USED DESIGN PATTERNS - SINGLETON

- ◉ We might utilize singleton pattern while we intend to have one and only one Machine per NewGame.