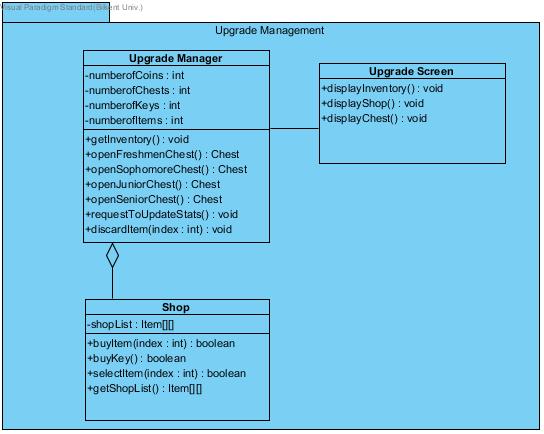
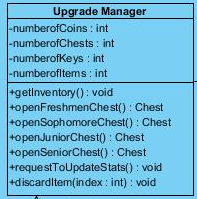
**Upgrade Management Subsystem**

Upgrade Management Subsystem is one of the major subsystems in the design process. This subsystem includes Upgrade Manager, Upgrade Screen and Shop classes. These classes have following functions;

* Upgrade Manager class handles the situation of player’s items.
* Upgrade Screen class displays the changes into the main screen.
* Shop class handles the purchase and removal of the items



**Upgrade Manager Class**

****

#### Attributes:

private int credits: It keeps the coin number that user has

private int numberOChests: It keeps the chest number that user has

private int numberOKeys: It keeps the key number that user has

#### Methods:

**public void getInventory():** This method gets player’s inventory. In other words, it shows how many keys, chests, coins does player has. It does not return anything.

**public Chest openFreshmenChest():**This method opens freshmen type of chest. To open freshmen chest, it is requires 1 key. It has 90% chance of giving a standard tier item, 7% chance of giving a rare tier item and 3% chance of giving an ultra-rare tier item. Therefore, according to these possibilities, this method gives the item and returns it.

**public Chest openSophomoreChest():**This method opens sophomore type of chest. To open sophomore chest, it is requires 2 key. It has 50% chance of giving a standard tier item, 30% chance of giving a rare tier item and 20% chance of giving an ultra-rare tier item. Therefore, according to these possibilities, this method gives the item and returns it.

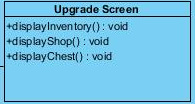
**public Chest openJuniorChest():**This method opens junior type of chest. To open junior chest, it is requires 3 key. It has 30% chance of giving a standard tier item, 35% chance of giving a rare tier item, 25% chance of giving an ultra-rare tier item and 10% chance of giving a “hacker” tier item.. Therefore, according to these possibilities, this method gives the item and returns it.

**public Chest openSeniorChest():**This method opens junior type of chest. To open junior chest, it is requires 3 key. It has 18% chance of giving a standard tier item, 25% chance of giving a rare tier item, 32% chance of giving an ultra-rare tier item and 25% chance of giving a “hacker” tier item.. Therefore, according to these possibilities, this method gives the item and returns it.

**public void requestToUpdateStats():**This method sends an upgrade request to game master in order to update the player’s stats. It does not return anything.

**public int discardItem(index : int):**This method is removes the item that right clicks on it and returns the index of this item.

**Upgrade Screen Class**

****

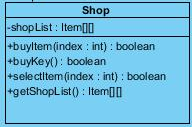
#### Methods:

**public void displayInventory():**This method displays the inventory of the player on the screen. It does not return anything.

**public void displayShop():** This method displays the shop on the screen in order to show the purchasable items from the shop. It does not return anything.

**public void displayChest():**This method displays the chests of the player on the screen. It does not return anything.

**Shop Class**

****

#### Attributes:

**private Item[] shopList:** It keeps the shop items into the one dimensional array with its index.

#### Methods:

**public boolean buyItem(index : item**): This method buys the item and it returns true if the item is successfully bought or false if the item is not bought.

**public boolean buyKey():**This method buys the key in order to open a chest and it returns true if the key is successfully bought or false if the key is not bought.

**public boolean selectItem(index : item):** This method selects the item whose index taken in the parameter and and it returns true if the item is successfully selected or false if the item is not selected.

**public Item[] getShopList():**This method gets the shop item’s list and returns them.

# 1. Introduction

## 1.1 Purpose of the system

Survival in Bilkent is a top to down 2D shooter game. The purpose of the game is to entertain the player. In order to design more enjoyable game, we created a variety of enemies such as bugs, assignments, quizzes, labs, midterms, and finals. These are the obstacles that the players shoot. Additionally, we designed our game that includes 4 levels. These levels’ difficulty increases from first to last level. Because of that, the game is more enjoyable and challenging.

## 1.2 Design Goals

### End User Criteria:

**User-Friendliness:** The game can be played easily. In other words, the mechanics are generic and self-explanatory that seen in many 2-D shooter games. Additionally, we create an understandable user interface.

### Maintenance Criteria:

**Extensibility:** Reusability and extendibility are crucial for software projects.Especially, we will add new upgrade items to the project. To addition,Survival in Bilkent can be modified and re-used in another project easily.

**Portability:** Our game will work in different software environments and we will implement our project in Java because, it has JVM that provides an opportunity to work correctly.

### Performance Criteria:

### Game Performance: The game should moves instantly. Average FPS should be greater than 30. Otherwise, the game cannot be playable and enjoyable.

### Trade Offs:

#### Performance vs. Memory:

#### Performance is important for our games. We want our game will run quickly. In order to moves our game smoothly, we should increase the memory space to give better game experience.

**Efficiency – Reusability:** We indicated that our game will be reusable. We will try to write the code as much reusable we can However, if the game’s efficiency is effected in a bad way, we will try to write the code more efficiently rather than reusability.

## 1.3 Definitions, acronyms, and abbreviations

MVC: Model View Controller

JDK: JavaDevelopment Kit

JVM: Java Virtual Machine

FPS: frames per second

## 1.4. References