DOKUZ EYLÜL UNIVERSITY ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 2210

Object Oriented Analysis and Design

INTERSTELLAR BORDER CONTROL SYSTEM

by

2020510086 Celal ÖZDEMİR

2019510042 Gufran GÜNEŞ

2019510050 Selim Eren KARAR

CHAPTER ONE

INTRODUCTION

1.1 Problem

Imagine a planet without a customs officer. Everyone would go in as they wanted and this would cause a lot of problems like murder, theft, incompatibility problems, cultural conflict etc. And imagine a customs officer without our software they would have to do tons of paperwork. Our software solves this problem and makes customs officers work easier

1.2 Goals

Our software has a user-friendly interface which customs officers can interact with ease. A customs officer can easily check the visitors identity and attributes with our program and judge accordingly. Our software makes it easy to keep both the visitors and the residents of the planet safe and sound.

1.3 Stakeholders

Several different types of stakeholders can be noted when it comes to our software. The most obvious ones are rulers of planets, countries and galactic federations. They might like to use our software on their borders to keep their countries safer.

1.4 Motivation

To make the Gotham Galaxy safer, better and more livable. We aimed to make a border control system interface that inspired from the video game “Papers Please”.

CHAPTER TWO

REQUIREMENTS

Application interface :

- Choose from available planets.
- View attributes of the visitor.
- View planets conditions.
- Choose to accept or deny the visitor in to the planet.
- Check current news in the planet.

Background operations:

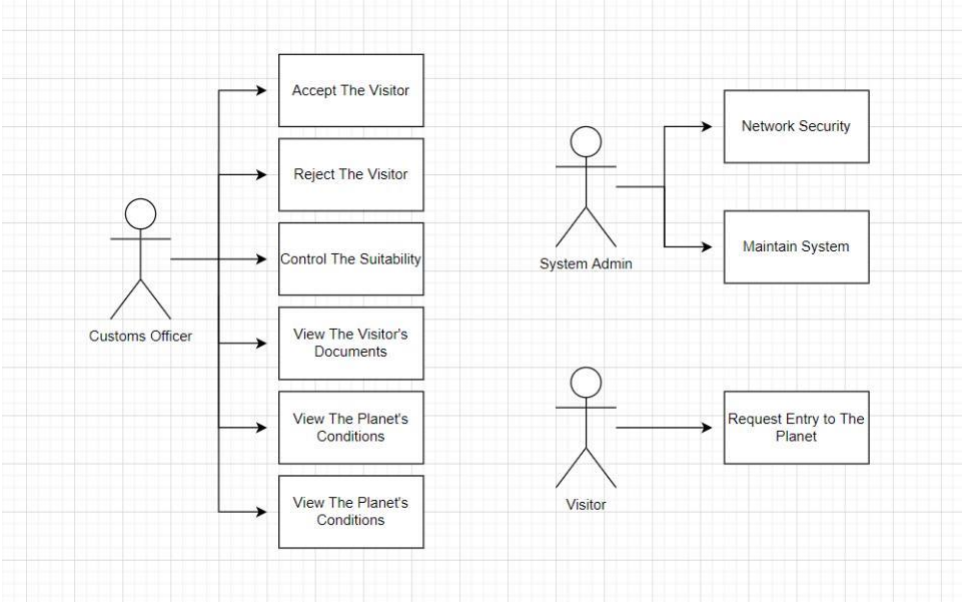
- Control operations.
- Determining the consequences of the decisions.
- Generating random aliens.

CHAPTER THREE

UML DIAGRAMS

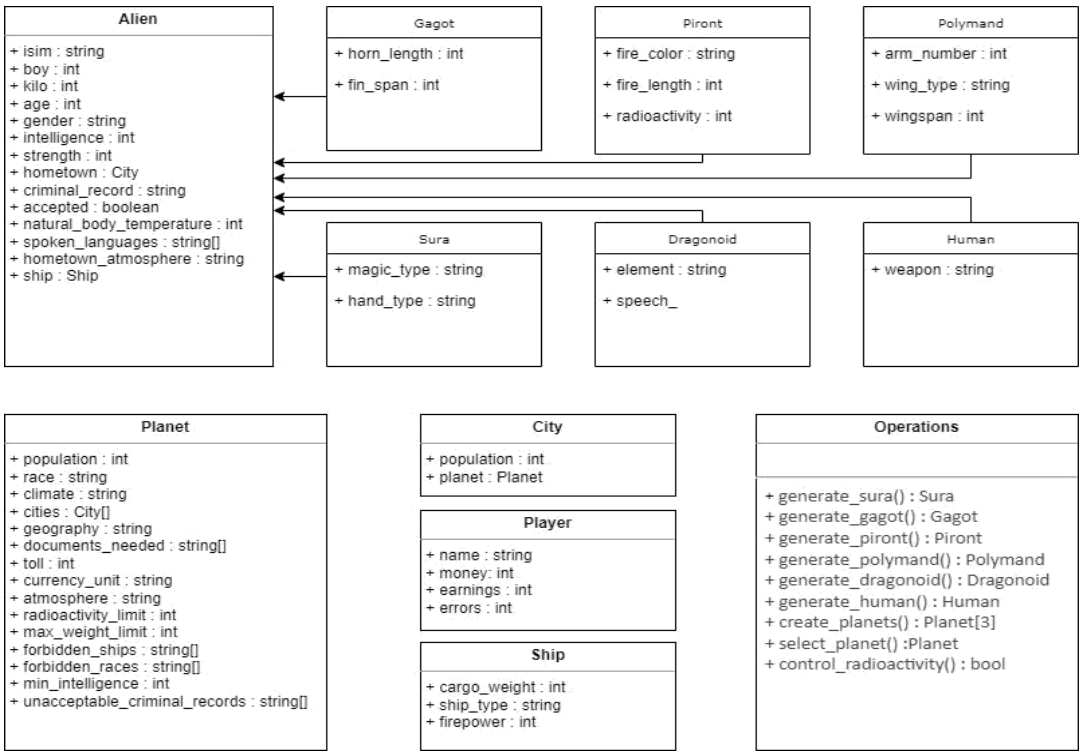
USE CASE DİAGRAM

Since we have only one main character due to the structure of the project, our use case diagram is like this. As drawn below, the actions that our officer can perform are clear. Apart from that, there is a system admin running in the background.



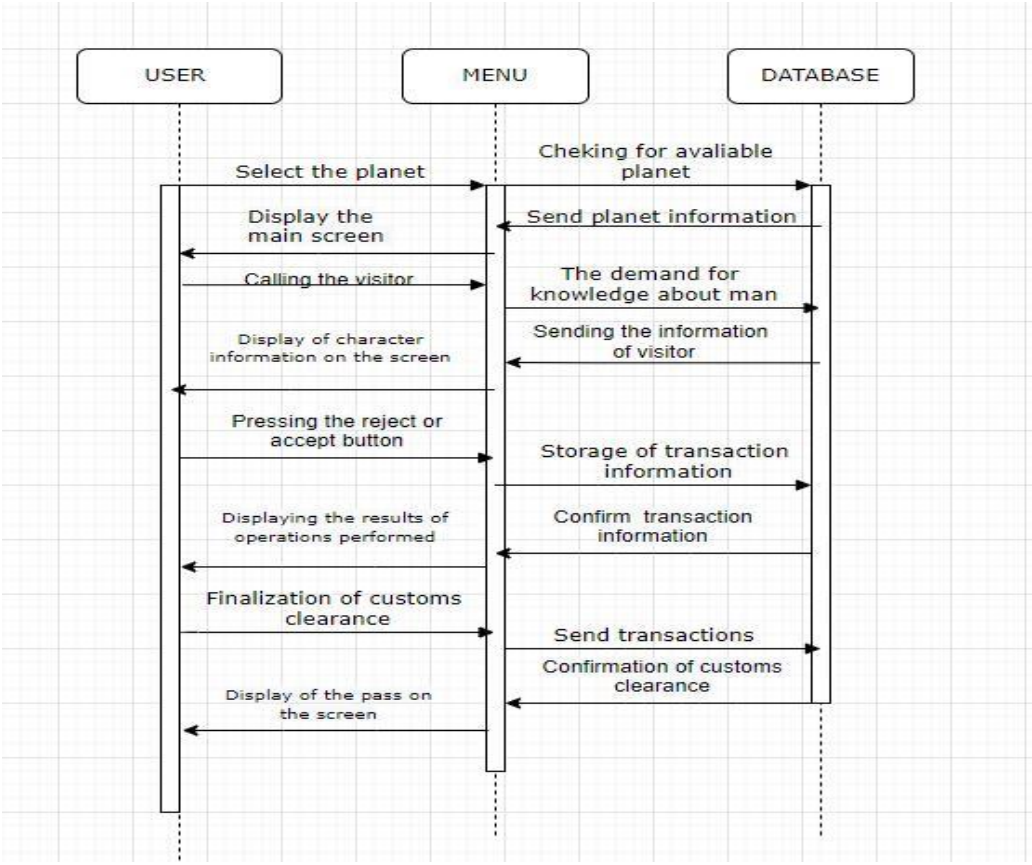
CLASS DİAGRAM

The class structures we have in mind for now are as follows. Of course, as the project progresses, there will be changes in our class structures. The current one is just a plan.



SEQUENCE DİAGRAM

In this diagram, we tried to briefly explain the events that took place on the main screen and in the background.



CHAPTER FOUR

IMPLEMENTATION

We started the coding part by creating the classes and variables according to the first 2 phases of the homework. Then we created couple of simple methods and functions to create objects which are connected to each other. Then we made methods that goes through all the objects and control their attributes. This basically created the basis of our game. Then we started to create the GUI of the game. Firstly, we created a main screen and made other windows that connected to the main window. These other windows have their own different styles, and they represent different planets in the game. We made the GUI look nice and the gameplay easy to learn and play.

Now to explain in detail our code Works in a loop that resides in the main method. In this loop we have 3 if statements that separates different planets. In every if statement program creates a random alien and prints the aliens' attributes to the screen. Then the program controls if the alien can enter the planet or not to check if the player made the right choice. Then the program rewards or warns the user according to the user's choice. This was basically how the main loop Works. We have multiple different functions to create planets, aliens, cities etc. We decided the attributes of the objects all by ourselves and created the manually. Program also have multiple different methods for checking if the different alien species can enter the planet or not. These methods check the aliens' attributes one by one using multiple if conditions then create errors which explains the reason why the alien can't enter the planet.

Now to talk about the GUI. We created a main window which user can choose between 3 planets and play the one they want. In our planet windows we have 4 methods that prints the text we sent from the main class to the screen. We created multiple JTextArea objects to print the information to the screen. Finally, there is 2 buttons in the planet windows that user can click to choose if they want to let the alien in or not. These buttons send the information as a boolean to the main class.

CHAPTER FIVE

Firstly, we learned how to make a game project from start to finish and experienced what difficulties we may face while doing so. We also learned how to do teamwork and time management. We learned that time management is important. We learned how to create an aesthetically pleasing GUI and code using OOP.

If we managed our time right, we planned to add music and difficulty options to our game. We also planned make the windows look better and we also planned to make different avatars for each alien that we created. We also were planning to make various planets, aliens etc. Another plan of ours was to have different outcomes of the choices that the player makes.