  
  
  
  
CS 319 - Object-Oriented Software Engineering  
System Design Report  
  
  
Animal Uprising  
Group 3-A  
Bora Ecer

Ata Gğn Öğün

Albjon Gjuzi

Tanay Akgül

[1. Introduction 3](#_Toc495951334)

[1.1 Purpose of the System 3](#_Toc495951335)

[1.2 Design Goals 3](#_Toc495951336)

[1.2.1 Efficiency 3](#_Toc495951337)

[1.2.2 Portability 3](#_Toc495951338)

[1.2.3 Reliability 3](#_Toc495951339)

[1.2.4 Extensibility 3](#_Toc495951340)

[1.2.5 Tradeoffs 4](#_Toc495951341)

[1.3 Definitions, acronyms and abbreviations 4](#_Toc495951342)

[1.4 References 4](#_Toc495951343)

[2. Software Architecture 4](#_Toc495951344)

[2.1 Overwiew 4](#_Toc495951345)

[2.2 Subsystem Decomposition 4](#_Toc495951346)

[2.3 Hardware/Software Mapping 4](#_Toc495951347)

[2.4 Persistent Data Management 4](#_Toc495951348)

[2.5 Access Control and Security 4](#_Toc495951349)

[2.6 Boundary Conditions 4](#_Toc495951350)

[3.Subsystem Services 4](#_Toc495951351)

[4. Glossary 4](#_Toc495951352)

# Introduction

## Purpose of the System

Animal Uprising is a 2D strategy/adventure game which aims to provide a well-designed and enjoyable gameplay to entertain the players. The gameplay experience starts with an easy level which will work as a tutorial for the players, so that they can learn the game. However, in order to provide more satisfaction and pleasure the next levels of the game is designed to be more challenging, so that the game will have the player’s attention, and increase the urge to play the game.

## 1.2 Design Goals

### 1.2.1 Efficiency

The main design goal of the system is efficiency, to achieve that, the system must be able to work in high performance. Since, a smooth gameplay is one of the most important feature which increases the player’s urge to play the game, we are going to minimize the memory usage and the CPU usage. In order to achieve that, first, we are going to implement the code in the most efficient way possible. Also, we are going to design the system so that the workload of the objects is going to be nearly balanced.

### 1.2.2 Portability

Since portability is an important feature for a software to have various users from different platforms, we have decided to implement the game in Java, since it provides platform independent software which will make our system portable through various platforms.

### 1.2.3 Reliability

Our system will be reliable in terms of being consistent with the boundary conditions. The system should not respond with any unexpected results -like bugs, crashes- which are not specified in the boundary conditions. In order to provide that, we are going to test the system in all possible ways during and after the development stage. Also, the boundary conditions will be selected carefully and with caution so that there won’t be a case with which puts the system in an unexpected situation. This will provide the system to foresee the possible fatal failures which will be dealt with.

### 1.2.4 Extensibility

In order to keep the interest of the players of Animal Uprising, the game requires to have new features and functionalities. For that the design of the game must be suitable to add further improvements and additions to the current system. In order to achieve this, object oriented architecture of our game must be designed in a way that each object should be able to operate with few dependencies. So that the modifications and further additions won’t cause any bugs or crashes.

### 1.2.5 Tradeoffs

* **Functionality vs Usability:**
* **Cost vs Robustness:**
* **Efficiency vs Portability:**
* **Rapid Development vs Functionality:**
* **Cost vs Reusability:**
* **Backward Compatibility vs Readability:**

## 1.3 Definitions, acronyms and abbreviations

## 1.4 References

# 2. Software Architecture

## 2.1 Overwiew

## 2.2 Subsystem Decomposition

## 2.3 Hardware/Software Mapping

## 2.4 Persistent Data Management

## 2.5 Access Control and Security

## 2.6 Boundary Conditions

# 3.Subsystem Services

# 4. Glossary