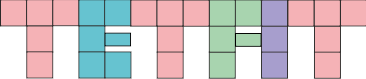


CS 319 - Object-Oriented Software Engineering

System Design Report



Group 1F

Irmak Akkuzuluoğlu

Irmak Türköz

Mustafa Culban

Yasin Erdoğdu

[**1. Introduction**](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.6zoolc1jb84v) 3

[1.1 Purpose of the system 3](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.f687jjezem0v)

[1.2 Design Goals 3](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.o5pjc4lm0k4l)

[Trade Offs: 4](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.y2569ztvrz5k)

[Development time vs Complexity 4](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.3viq5d87s6g7)

[Ease Of Use and Ease of Learning vs. Functionality: 4](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.13tkeqibum48)

[1.3 Definitions, acronyms, and abbreviations](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.ccsidqbuyjmm) 4

[Abbreviations:](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.rwtur0frkdf9) 4

[1.4. References](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.jijk7yp501un) 4

[1.5. Overview](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.67318devjuds) 4

[2.1. Overview](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.ccay0p4cuw3g) 5

[2.2. Subsystem Decomposition](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.rlsk19gkksif) 5

[2.3. Architectural Styles](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.r71f6rnb113p) 5

[2.3.1 Layers](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.tb3s211rruxm) 5

[2.3.2 Model View Controller](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.sdaphzav8i2q) 5

[2.3. Hardware / Software Mapping](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.qew20wfix1hn) 5

[2.4. Persistent Data Management](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.vltvj0174r80) 5

[2.5. Access Control and Security](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.33pckveebqtr) 5

[2.6. Boundary Conditions](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.88t6vmyuqvfb) 5

[**3. Subsystem  Services**](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.5e744hja5dph)  **5**

[3.1. Design Patterns](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.takypevfq4uq) 5

[3.2.  User Interface Subsystem Interface](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.wlssyuaiaih3) 5

[3.3 Game Management Subsystem Interface](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.qnyhsuovn74d) 5

[3.4 Game Entities Subsystem Interface](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.6kxctzpi5wbf) 5

[3.5 Detailed  System Design](https://docs.google.com/document/d/18yDRMottXQoQqH78ta_ycetd7yZK1Xl1ZT-r74xdzUM/edit#heading=h.pvsmqy4buxfd) 5

**1. Introduction**

**1.1 Purpose of the system**

Tetfit is a 2-D tile matching puzzle game, which is intended to entertain users greatly. It has minimalistic graphics, consisting of simple shapes and a user-friendly interface. The gameplay is easy to understand, but the game gets challenging as the player carries on with the game. Tetfit also has different levels of difficulty which gives the players the freedom to choose where to start from, making gameplay more individualistic and enjoyable. It is a skill-based game, it has two different modes that both triggers strategical thinking. Overall, it is a great pastime activity with a pleasing design.

**1.2 Design Goals**

Below are some important design goals of the system which are proposed to meet non functional requirements of our project, which were mentioned previously in the analysis report.

**Usability:** The ease of use is one of the most important aspects of the system for it is a game aiming to entertain people. In order to provide this, player should be able to get a grip of the system easily. For this purpose, we will provide the user friendly simplistic interfaces, so the user can achieve each and every operation he desires to perform. Moreover, it is decided that the menu operations will be done by mouse clicks and the game will be controlled with the arrow keys for they are the most common and easy ways to control a game.

**Ease of Learning:**  In order to let the user have fun without dealing with confusion we intended to give the game the ease of learning. Tetfit is overall a simplistic game, with an easy main goal. The information about the main goal and how to play the game will be provided by the help document which will can be accessed through the main menu. This documentation will be detailed enough to avoid any kind of confusion. Since the game is already an easy to understand game, we expect the users to learn and play the game easily.

**Extendibility:** Letting the game to extend and change is a crucial point to maintain the interest of the users in general, by adding new components, features to the game. In order to achieve this our system will be open to any kind of change, making it convenient for us to add new functionalities and entities easily to the present system.

**Modifiability**: Extendibility also requires modifiability. Modifiability provides the ease to modify the system to correct faults, improve performance or other attributes, or adapt to a changed environment. Throughout the implementation this specification will be kept in mind so that our system will allow us to improve it.

**Portability:**  This is also an important issue for a software system, for it can allow a  wide range of users to access it. With this consideration, we decided to implement our system in such manner. Our game will be implemented in Java, because its JVM maintains platform independency, allowing the system to achieve the portability.

**Response Time:** For our project is a game, another crucial goal is to keep a short response time. It is vital to provide quick responses to the users’ requests, in order to maintain a pleasing experience for the users. Our goal is for the system to respond to users’ actions as quick as possible, while also smoothly displaying animations and screening the gameplay.

**Trade Offs:**

1. **Development time vs Complexity**

For our system has to be completed by the end of this semester, we have a quite limited time to develop and implement our project. We are trying to optimize the usage of time to achieve as much as possible, but still there needs to be a trade off between the development time and the complexity. Our purpose is to provide a high quality game but with a limited amount of variety in the gameplay. We limited our game to 2 different mods which will have different difficulties.

1. **Ease Of Use and Ease of Learning vs. Functionality:**

Our system aims to provide an ease of understanding and usability. Hence, we need to trade off the functionality to keep the game simple. This is planned to be achieved by avoiding complex functions and by not exceeding a certain number of different functional abilities that will be provided to the user. We will keep our minimalistic view, and implement the best possible game which easy to understand and play by sacrificing a bit of the functionality.

1. **Performance vs. Memory:**

In our game, it is vital to provide the player smooth animations, effects, transitions, for an enjoyable gameplay. thus, performance is the primary focus of the system. However, in order to achieve this we had to sacrifice the memory.

**1.3 Definitions, acronyms, and abbreviations**

**Abbreviations:**

**1.4. References**

[1]<http://en.wikipedia.org/wiki/Java_(programming_language)>

[2] <http://www.nada.kth.se/~karlm/prutt05/lectures/prutt05_lec7.pdf>

[3] <https://drive.google.com/file/d/0B9ApNnKlfgcHeUxKbFAyQTNQbE0/view>

[4] <https://drive.google.com/file/d/0B9ApNnKlfgcHYllRbEZEYjFRRzg/view>

[5] <http://www.ieee.org.ar/downloads/Barbacci-05-notas1.pdf>

**1.5. Overview**

This section, generally summed up the main purpose of the system, that is to provide entertainment to users. In order to meet every intended goal, we defined them clearly, in detail. Our main focus on design goals are usability, ease of learning, extendibility, portability, modifiability and high performance. We are aware that every system requires some trade offs to be made, hence we planned these trade offs to be as favourable as possible for our project. We made some trade offs between development time and complexity, ease of use and functionality, lastly performance and memory.

**2. Software Architecture**

**2.1. Overview**

               The purpose of section to decompose our system into maintainable subsystems. Our software architecture aims to have reduced coupling which measures the dependencies between two subsystems and increased cohesion which measures dependencies among classes within a subsystem.

**2.2. Subsystem Decomposition**

**2.3. Architectural Styles**

**2.3.1 Layers**

**2.3.2 Model View Controller**

**2.3. Hardware / Software Mapping**

**2.4. Persistent Data Management**

**2.5. Access Control and Security**

**2.6. Boundary Conditions**

**3. Subsystem  Services**

**3.1. Design Patterns**

**3.2. User Interface Subsystem Interface**

**3.3 Game Management Subsystem Interface**

**3.4 Game Entities Subsystem Interface**

**3.5 Detailed  System Design**