

UNIVERSITY OF APPLIED SCIENCES
ASCHAFFENBURG

Software Design Specification

Project: Development of a Biofeedback Application

Phase: Design

Authors:
Kerim Turacan

Documentname: AB-BiofeedbackApplication-DS-3

Version: 1.2

Creation date: 29.05.2020

File: BiofeedbackApplication-DS.pdf

Modifications – Document Status

Version	Status	Creation Date	Editor	Modifications
1.0	Planned	29.05.20	Kerim Turacan	Initial Document
1.1	Under Construction	31.05.20	Kerim Turacan	Adding and Finishing
1.2	Presented	02.06.20	Kerim Turacan	Minor Changes

(Status ::= planned, under construction, presented, accepted)

TABLE OF CONTENTS

Software Design Specification.....	1
1 Introduction.....	4
1.1 Purpose.....	4
1.2 Definitions, Acronyms and Abbreviations	4
1.3 References.....	4
1.4 Overview.....	4
2 System Overview.....	5
2.1 Functionality.....	5
2.2 System design.....	5
3 Design Considerations.....	5
3.1 Assumptions and Dependencies	5
3.2 General Constraints.....	5
3.3 Goals and Guidelines.....	5
3.4 Development Methods.....	5
4 System Architecture.....	6
4.1 GUI layer	6
4.2 Logical layer.....	7
4.3 Data Layer.....	7
5 Detailed System Design.....	8
5.1 GUI Layer.....	9
5.2 Class diagram of the Logical Layer.....	9

1 Introduction

1.1 Purpose

The purpose of this document is to explicitly describe the software design of the product. The document is based on the System Requirement Specification of this project [AB-BiofeedbackApplication-SRS-1].

1.2 Definitions, Acronyms and Abbreviations

GUI	Graphical User Interface
HR	Heart rate

1.3 References

[SRS]	Software Requirements Specification Version 1.2
[AM]	Analysis Model Version 1.2

1.4 Overview

Section 2 of this document presents an overview of the system. Section 3 describes design considerations. The system architecture is explained in Section 4. The detailed system design with all system components is described in Section 5.

2 System Overview

2.1 Fuctionality

Scope of this Software is to track the customers heartrate while playing the main game “Space Invaders” beside some implemented minigames. Meanwhile graphical, auditive and difficulty changes during the game are placed in predefined timesteps to stimulate/effect the players in-game behavior and its heartrate.

2.2 System Design

The architecture concept is a classical 3-layer-design, which means the software can be separated in a presentation-, a logical- and a data-layer. For a more specified overview see section 5.

3 Design Considerations

3.1 Assumptions and Dependencies

Before using this software, you must install Java on an operating system/ platform of your favor (recommended: Microsoft Windows 10) to run applets. The measuring input sensor for heartrate data is the “JOY-IT DEBO SENS HEART“ which is used on an Arduino Board.

3.2 General Constraints

As mentioned in 3.1.

3.3 Goals and Guidelines

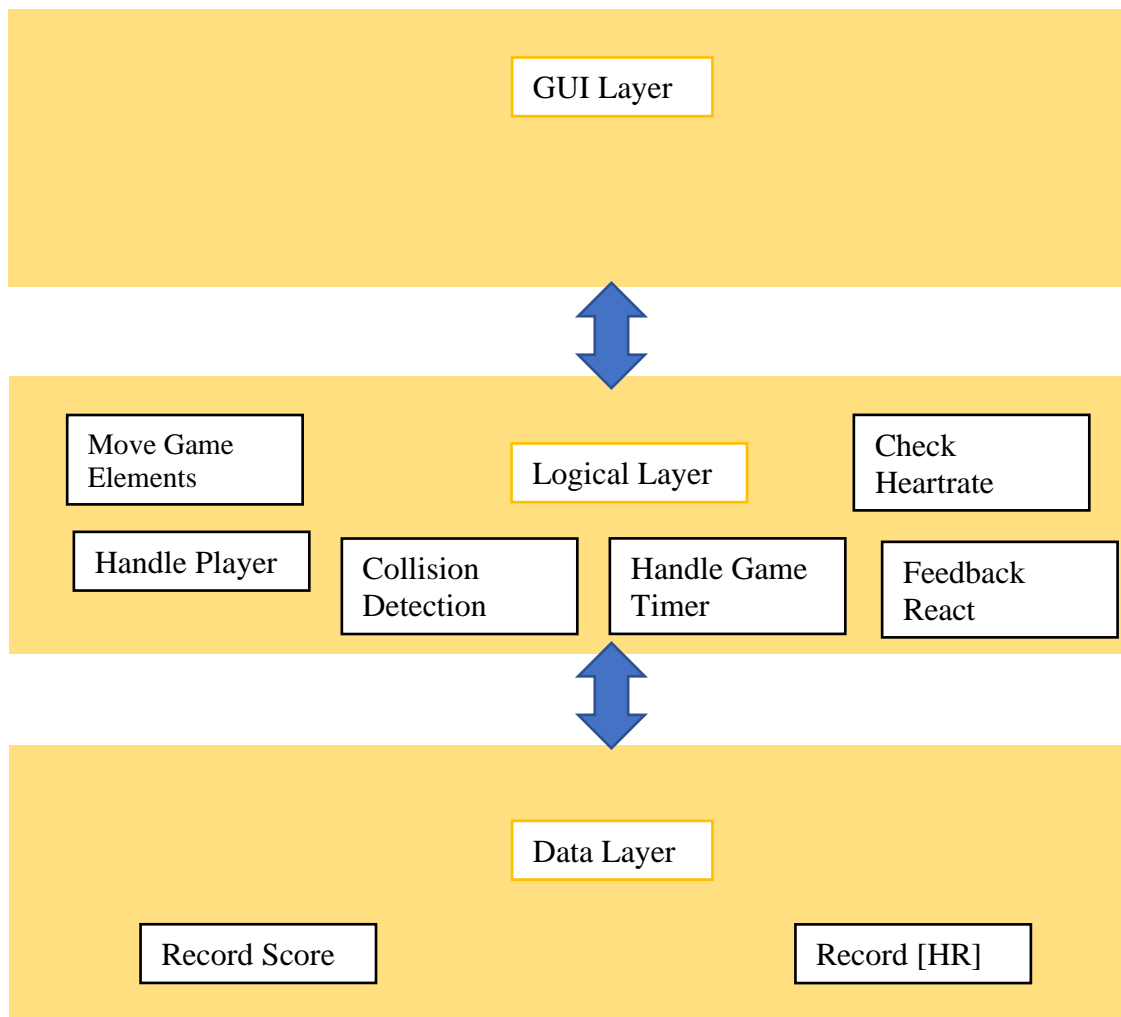
A major task is to make the code fast and easy to maintain. The GUI of the software will be done with JavaFX, which allows to create Java applications with a modern, hardware-accelerated user interface that is highly portable.

3.4 Development Environment

The following tools were used in development process

- | | |
|-------------------|---|
| • JDK | Java 14.01 |
| • IDE | Eclipse IDE for Java Developers – 2020-03 |
| • SDK | JavaFX 14 |
| • Scene Builder | SceneBuilder 11.0.0 |
| • Database (DBMS) | Firebird |
| • SQL client | DBeaver |

4 System Architecture



The Figure above shows an overview of the Biofeedback Application. It is a classical three-tier-system, with a GUI-, a logical- and a data-layer. As shown in this figure, the application is divided into smaller subsystems. Parts of these subsystems are almost independent from the other parts.

4.1 GUI Layer

The main menu GUI is split up in different scenes realized with the eclipse tool “Scene Builder”. The first scene is to optionally enter in customers data to combine heartrate sensor data with the players age and weight, for an easier analyzation. After entering the data in TextFields and pressing the “NEXT”- Button, the next scene is showed which maintains the “START”-button, to start the game whenever the player is ready.

The actual game contents are split up in two different areas. The main game “Space Invaders” is displayed on the left side with 70% window margin. On the right-hand side at the top, a score label shows the current game score while underneath the minigames are placed.

4.2 Logical Layer

- **Move Game Elements:** After creating the game elements in the Stage initialization the game elements (Enemies), namely Meteors are calculated to constantly move downwards.
- **Handle Player:** With Key Inputs the player can be moved to the left and right and is able to shoot bullets.
- **Collision Detection:** This subsystem is responsible to detect every collision of any objects with another and updates its position and the game score.
- **Handle Game Timer:** In the background, the game timer is responsible for time critical actions. It is used for updating the game score and calling the minigames at a specific time.
- **Check Heartrate:** Constantly tracks the heartrate of the user.
- **Feedback React:** This subsystem changes the whole user experience of this application. By triggering visual, auditive effects, as well as difficulty changes the user should change his in-game behavior which should lead to heartrate changes, which is the target of changing the game mode.

4.3 Data Layer

The heartrate data and the score progress are recorded and saved on the database.

5 Detailed System Design

5.1 GUI Layer

Since this layer is implemented using JavaFX and Scene Builder, no further details need to be provided.

MainMenu.fxml:

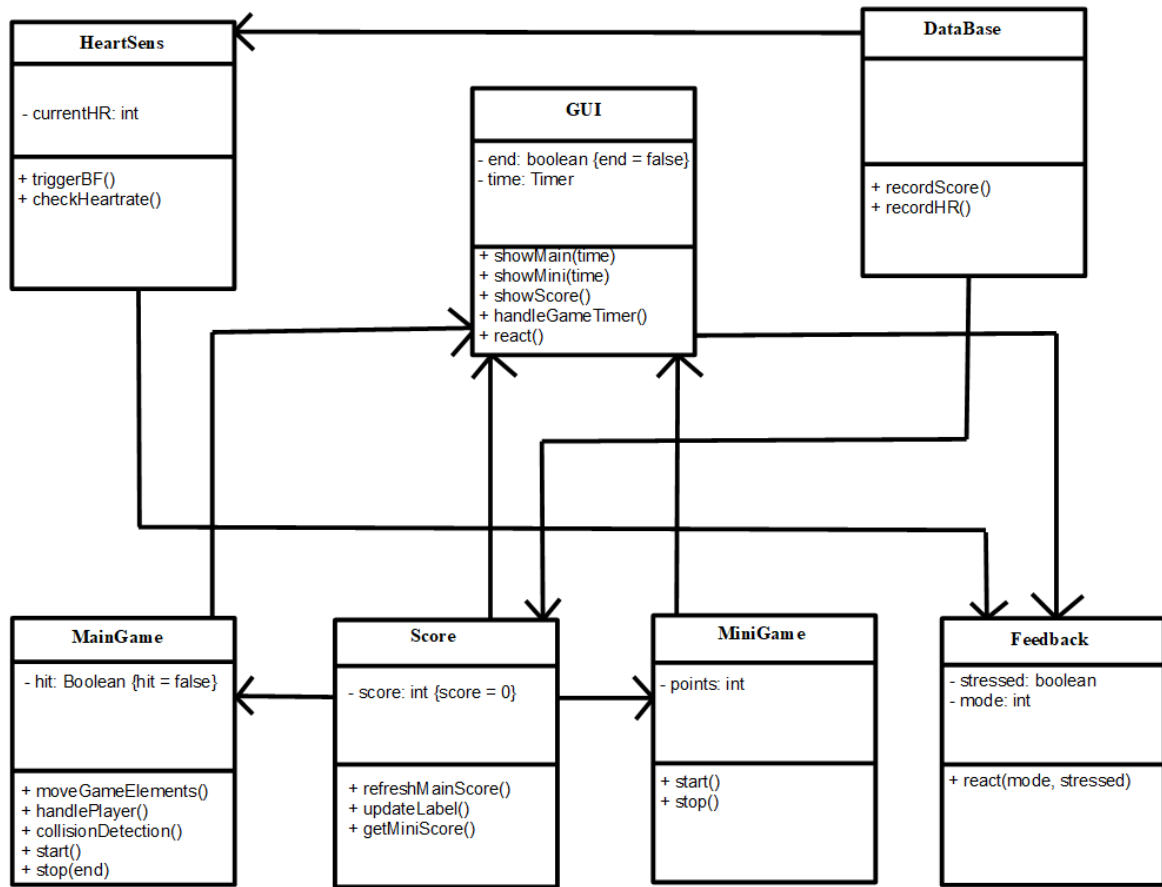
```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="369.0" prefWidth="497.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="view.MainMenuController">
    <children>
        <ImageView fitHeight="112.0" fitWidth="154.0" layoutX="340.0" layoutY="23.0"
pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@../../../../PNG/logo/th_ab_logo.png" />
            </image>
        </ImageView>
        <Button alignment="CENTER" focusTraversable="false" layoutX="104.0"
layoutY="155.0" mnemonicParsing="false" onAction="#StartButtonPushed"
prefHeight="29.0" prefWidth="74.0" text="START" />
    </children>
</AnchorPane>
```


5.2 Class diagram of the Logical Layer

The following figure shows a class-diagram of the logical layer of the Biofeedback Application.



The classes and methods will be documented in detail in the code. The terms of the methods are self-explanatory and therefore not documented in detail.