UNIVERSITY OF APPLIED SCIENCES
ASCHAFFENBURG

Analysis Model

Project: Development of a Biofeedback Application

Phase: Requirements Analysis

Authors:
Tim Schmitt

Documentname: AB-BiofeedbackApplication-AM-2

Version: 1.2

Creation date: 14.05.2020

File: BiofeedbackApplicationAM.pdf

Modifications – Document Status

Version	Status	Creation Date	Editor	Modifications
1.0	Planned	14.05.2020	Tim Schmitt	Initial Document
1.1	Under Construction	16.05.2020	Tim Schmitt	Adding and Finishing
1.2	Presented	18.05.2020	Tim Schmitt	Minor Changes

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

TABLE OF CONTENTS

1 Introduction

The following document is part of the Analysis Model and explains sequences of our program with use-case and class diagrams. With these diagrams you get gradually an exact overview of our program.

1.1 Purpose

The purpose of this document is to complete the analysis of the biofeedback application and show the connections in the program with diagrams.

1.2 Scope

The biofeedback application will test the effects of auditive and visual reactions to the user's heart rate by playing stressful games while monitoring his pulse. The user will play games simultaneously to further put him in stressful situations. The results of different runs will be evaluated by the software and stored in a database.

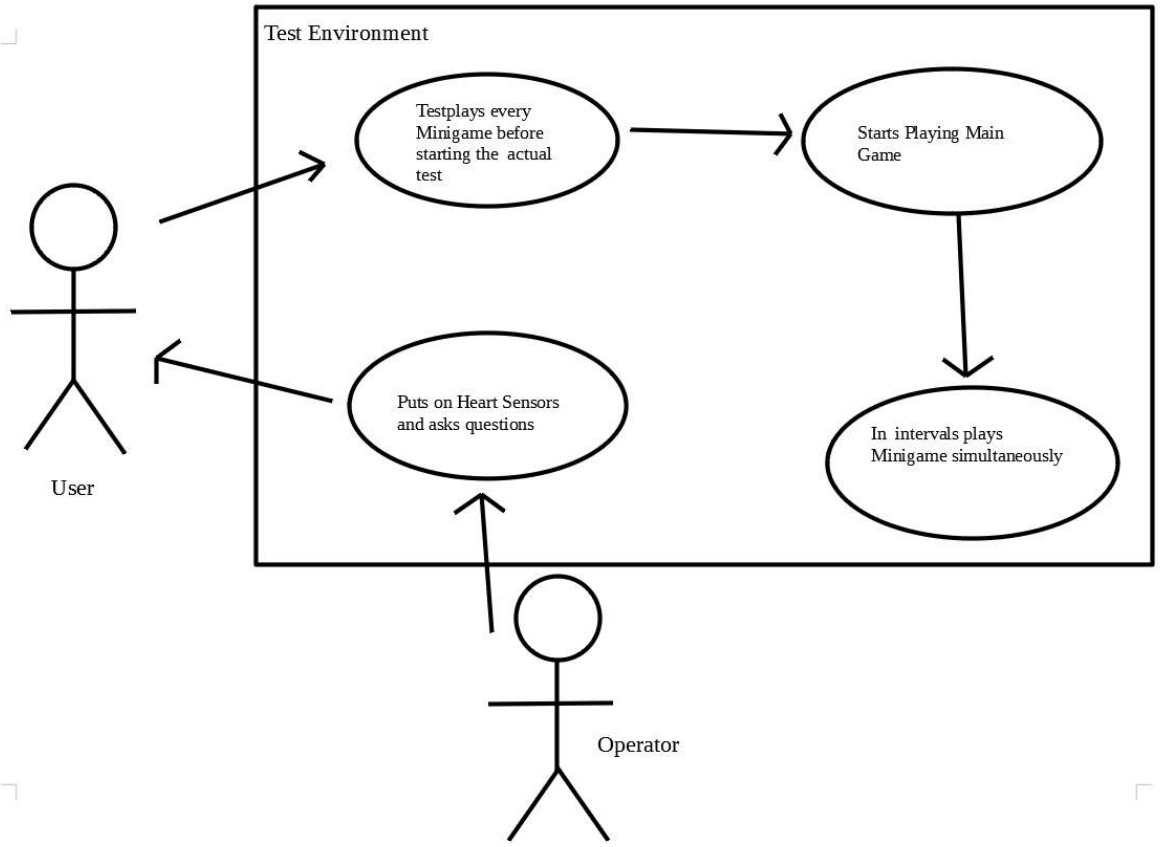
1.3 Definition, Acronyms and Abbreviations

GUI	Graphical User Interface
HRS	Heart Rate Sensor

1.4 Overview

Section 2 provides an object oriented analysis of the entire project. It contains a use case diagram.

2 Object-Oriented Analysis



2.1 Use Case Diagram

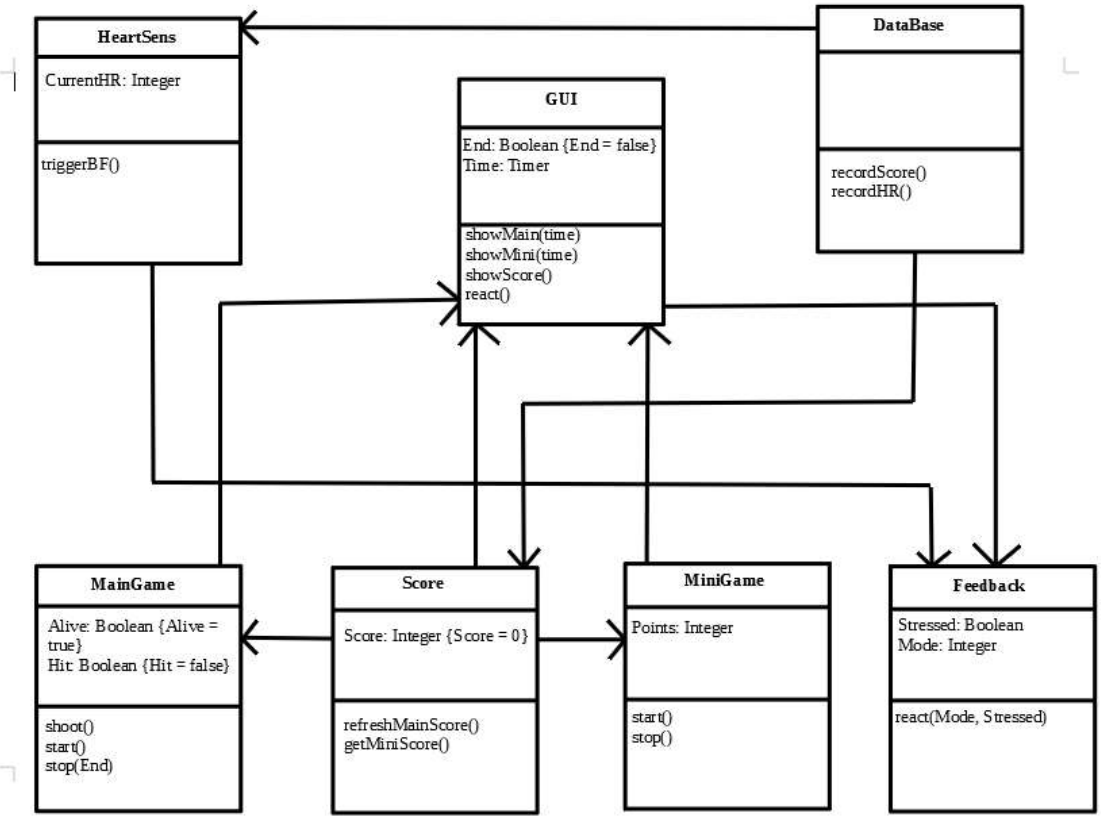
Actors: User, Operator
Activator: Operator starts test
Results: A user plays a game while simultaneously being faced with mini games to test activating or deactivating biofeedback.

User options:

- Before starting the game the user is connected to two heart rate sensors to better monitor his heart rate. Then he answers questions necessary for the test.
- The user has the option to test-play the mini games he is faced with during the run to get used to the controls.
- The user then has the option to start the game by pressing the start button.
- In the main game the user can either press a key to dodge left or a key to dodge right. He also can press a key to shoot.
- When faced with a mini game the user presses different keys to play while still having to control the main game
- At every time the user can see his score

2.2 Static Model

2.2.1 Class Model



2.2.2 Class Descriptions

HeartSens

Purpose:

Contains the current Heart Rate measured by the HRS connected to the users finger/ wrist. Depending on the heart rate it triggers the biofeedback.

Attributes:

- CurrentHR
  - description — the current heart rate is the mathematical average of both HRSs measured pulse
  - type — Integer

Operations:

- triggerBF
  - description — if the heart rate passes a certain value the function triggers the stressed attribute in the feedback class
  - arguments — none
  - return value — none

DataBase

Purpose:

DataBase class Is used to record the score and the heart rate during the test and save the data into an external database.

Attributes:

Operations:

- recordScore
  - description — records the score equivalent to the HRS's update frequency and saves it into the database
  - arguments — none
  - return value — none

- recordHR
  - description — records the heart rate measured by the sensor according to the sensors update frequency and saves it into the database
  - arguments — none
  - return value — none

MainGame

Purpose:

The main game is a space invaders type of game which lets you control a spaceship. It is played during the whole run.

Attributes:

- Alive
  - description — tells you if the player is alive or has hit a rock which will later determine the amount of points the player gets
  - type — boolean
- Hit
  - description — this value is set to true if the player shot an object with the blaster
  - type — boolean

Operations:

- shoot
  - descriptions — when the user presses space bar the blaster shoots a laser beam from the front of the spaceship
  - arguments — none
  - return value — none
- start
  - description — starts the game and the Timer in the GUI class
  - arguments — none
  - return value — none
- stop
  - description — ends the game after the time runs out which is indicated by the End value
  - arguments — End is the value that is set to true if the timer reached the end time
  - return value — none

MiniGame

Purpose

The mini game is supposed to stress the user by multitasking. There are several mini games played during the test run that each require different key inputs. You get extra points for every mini game played successfully.

Attributes:

- Points
  - description — the amount of points you get for completing the mini game (set by the operators in advance)
  - type — Integer

Operations:

- start
  - description — starts the mini game
  - arguments — none
  - return value — none

- stop
  - description — ends the mini game if finished or a certain amount of time ran out
  - arguments — none
  - return value — none

Score

Purpose:

The longer the player did not hit an object the more points he gets playing the main game. He gets additional points for completing a mini game. If he hits a meteor the score decreases. The score class contains the current score of the test.

Attributes:

- Score
  - description The current score during the test (starts at 0)
  - type Integer

Operations:

- refreshMainScore
  - description Checks if the player is alive and adds points if he is. If he hit a rock the points decrease. The longer he no death happened the more points he gets.
  - argument none
  - return value none

- getMiniScore
  - description if a mini game is finished the operations takes the points gained from the mini game and adds them to the current score
  - arguments none
  - return value Integer

Feedback

Purpose:

Feedback class contains the biofeedback implementations and is used to react to stress created by the games.

Attributes:

- Stressed
  - description — the HRS triggerBF function sets this value which determines whether the biofeedback starts or not
  - type — boolean
- Mode
  - description — Mode is set by the operators and determines which type of biofeedback is used in this run. (Activating = 1, No Feedback = 0, Counteracting = -1)
  - type — Integer

Operations:

- react
  - Description — starts the implemented biofeedback methods according to stress level and game mode. Only starts if Stressed is set to true and only uses biofeedback methods declared in that game mode. Also triggers react function of GUI class.
  - arguments — Mode, Stressed
  - return value — none

GUI

Purpose:

The GUI class is used to display everything the user sees on the screen. It also determines if the game ends and displays the biofeedback.

Attributes:

- End
  - description — Is set to false by default and only is true if the time, defined by the operators, is over.
  - type — boolean
- Time
  - description — Time is a timer that is used to check if the game time is over or not and to trigger different operations
  - type — Timer

Operations:

- showMain
  - description — displays main game on the screen and starts the timer
  - argument — time
  - return value — none
- showMini
  - description — displays mini game on the screen at certain points in the game time
  - argument — time
  - return value — none
- showScore
  - description — displays the current score on the screen and each gain or loss
  - argument — none
  - return value — none
- react
  - description — is triggered by the react function of Feedback class and displays the changes on the screen
  - argument — none
  - return value — none

2.3 Dynamic Model (Dynamisches Modell)