

# Contents

<b>1</b>	<b>Problems and solutions Highlights</b>	<b>3</b>
1.1	Avatar . . . . .	3
1.2	Visual studio Express . . . . .	3
1.3	GUI Testing . . . . .	4
1.4	File uploading . . . . .	4
1.5	GitHub . . . . .	4

# Chapter 1

## Problems and solutions Highlights

During the project several problems arose that needed solving. This chapter will highlight some of those problems and their solutions. The problems that will be discussed in this chapter will be the avatar, file uploading to the database, displaying graphs, testing the gui and using the tools visual studio and github. These problems will be discussed chronological.

### 1.1 Avatar

One of the main goals of the project was to use an avatar to stimulate a patient with his/her therapy. There was no need to program this avatar, because an implemented avatar was already given. But this caused other problems because the given implementation reduced the flexibility to use the avatar.

The first problem with the avatar was to get it displayed inside one of the program's windows. To achieve this, the process of the avatar must be started separately and the process and window handle must be retrieved. The problem that arose was that rescaling of the avatar didn't always succeed, neither did linking the avatar window to a window of the program.

The solution that has been found to this problem is not a neat solution. For example, timers are used to time the correct moment for rescaling the avatar because no clear point can be determined to do this. This causes of course that slower computers do not display the avatar correctly. Another difficulty was that you don't want the avatar process to close when the avatar is not displayed anymore, because loading of the avatar takes a long time. The window of the avatar needs to be hidden and displayed again when the avatar is needed again. The solution has been refactored several times to increase the efficiency of the use of code because multiple GUI elements use the avatar. Still a problem is that the implementation is rather specifically based on the avatar process that is used and cannot be changed easily.

The second problem with the avatar was and still partly is, to let the avatar show emotion and to let him talk. One of the problems is the robustness of the avatar itself and it still is. Quite randomly the avatar does or does not respond to the commands. Sometimes the avatar stops responding and it is not able to respond to multiple consecutive calls. Another problem is that the first time a command is given, it takes the avatar a long time to react, varying from 4 to 20 seconds. No solution for these problems has been found yet.

### 1.2 Visual studio Express

For the project, we needed to code in the language C#, because one of TA's had created a library with C# for connecting to the avatar, that was written in python. Visual Studio Express 2010 was decided to be the IDE in which the team would code, because it is the easiest IDE for developing in C#. When the coding began, all went well for the first couple of weeks. But then the team faced the limitations of the free version of Visual Studio. We couldn't use a database plug-in, there were no to-do lists and we couldn't use plug-ins in general. The team decided to use the ultimate version of Visual Studio, the

downside to this solution was that this is not free but it had an evaluation period, luckily the period would end after the context project had ended.

### 1.3 GUI Testing

One of the demands of the project was that it needed to be coded by the test-driven-development rules. That means that all code should be tested including GUIs. For standard Unit testing the team found that NUnit was quite helpful, but for explicit GUI testing no suitable tool was found that was compatible with Visual Studio Express. Because the team valued automated GUI testing very highly, the team decided it was worth the effort that an own testing suite should be developed. The implementation excessively uses reflection to communicate with the to be tested GUI, for verifying and button clicking etc. The downside of using reflection, it depends on the name of the component that is being tested. So code inspection is required and if the name of the component is changed, the test no longer works. There also the name has to be changed.

### 1.4 File uploading

When the team decided that the therapist could record messages for his/her patients, so that the avatar could talk to the patients, the problem was encountered that for this files needed to be uploaded to the database. It was needed to upload an audio file to the server and that it could be downloaded and played. The team uses an EWI SQL database server, so only text could be uploaded. So the first sub problem was to convert an audio file into plain text. Then it was discovered that the server has a 1mb upload limit, and the team found that the audio file would be too short. So the next sub problem was to split this string into multiple sections of maximum 1mb and upload it separately. All that was needed now was to download the multiple sections, merge them together and convert them back to the audio file that was recorded by the therapist so that the avatar can speak to the patient. The next problem was to let the avatar actually speak.

### 1.5 GitHub

GitHub is a tool that is developed to make it easier for a team develop code with each other. But in this case the team had little experience with github, so the use of github caused some problems. Lots of hours have been lost to solving problems with git, solving merge conflicts and trying to understand what's going on. Though the experience with github has increased during the project, still some problems keep coming back. Certain files (for example the project description file) cause merge conflicts each time a merge is done. Another example is that github integration with windows is not fully completed. For example a merging tool is not installed. Because no plugins can be installed with visual studio express (because of the limitations for the free version), github could also not be integrated in the coding IDE.