Software Engineering Project

- Bug World Specification -

P. Baumann, Constructor University, Spring 2023

Deadline: 2023-mar-02 23:59

Submission: PDF document per group, by email to TAs

1 Introduction

Task of this semester is to implement a game, the *Bug World*, in a Web browser.

In this phase, you need to

- Team up in groups of 4 – 5 people
- Jointly establish a specification of the game, based on the material provided.

Among the **skills trained** will, first and foremost, be programming of non-trivial software systems in changing teams. Further, immersion into new tools is an important exercise, plus several more – like teamwork, self-organisation and independent studies.

2 Task

Your task in this assignment is to devise a **system design specification** consisting of text, UML class and sequence diagrams, and any other suitable means enabling a programmer who has never heard of the game can implement it.

In addition to the specification, you should provide test cases containing examples of both good and bad input, and the expected system reaction (output).In other words: provide concrete test cases which will later on be applied to the implementations.

The following files are provided as your working basis:

- Simulator man page – This is the man page of the Bug World simulator. It is originally written for a command line implementation and, hence, is not applicable directly. Rather, the functionality described needs to be translated meaningfully to a Web environment.
- Assembler man page – This is the tool which translates bug programs from symbolic level (“assembler”) into the bug machine code which the simulator understands.

Read the description carefully, and ask any question necessary. Compared with real world specifications, this document is rather detailed and technical in order to get you started quickly. That said, the text provided is not necessarily complete - importantly, determining what’s missing and collecting rele-

vant information is **your** responsibility. In other words: You cannot say afterwards "we have not been told that".

If you find some piece of information undocumented in the material provided you may ask for clarification (through the channels indicated) or make reasonable assumptions. In any case, clearly list in your document (recommended: extra section) everything that goes beyond the documentation provided.

The **expected result** is a single PDF document per group. Clearly indicate the team members! Mail the PDF to the TAs before the deadline – a deadline is a deadline is a deadline.

Specifications are competitive, the best one will be selected as the basis for the following XP phases.

3 Hints

- Techniques for establishing such documents, and the required contents, are discussed in class.
- Questions can be asked in class or on the mailing list, course-advcs2@lists.jacobs-university.de, or in the helpdesk sessions. Not through individual mails.
- If you find relevant information elsewhere (and you are encouraged to seek!) you may use (short) text portions and screenshots as long as you cite appropriately, otherwise we have to file it under plagiarism. Longer text portions you have to paraphrase.

Some further thoughts:

- A good design can be adapted to changing requirements easily. Make sure your design is not overly restrictive.
- Try to achieve separation of concerns. The number of foreign interfaces used by a given module/ class should be small (loose coupling of classes).
- Make sure you use sensible and consistent names for classes, attributes, associations etc.
- Where necessary, you should explain design decisions.
- Focus on the overall design and avoid programming language specifics.

4 So…

let's roll and get our hands dirty! ☺