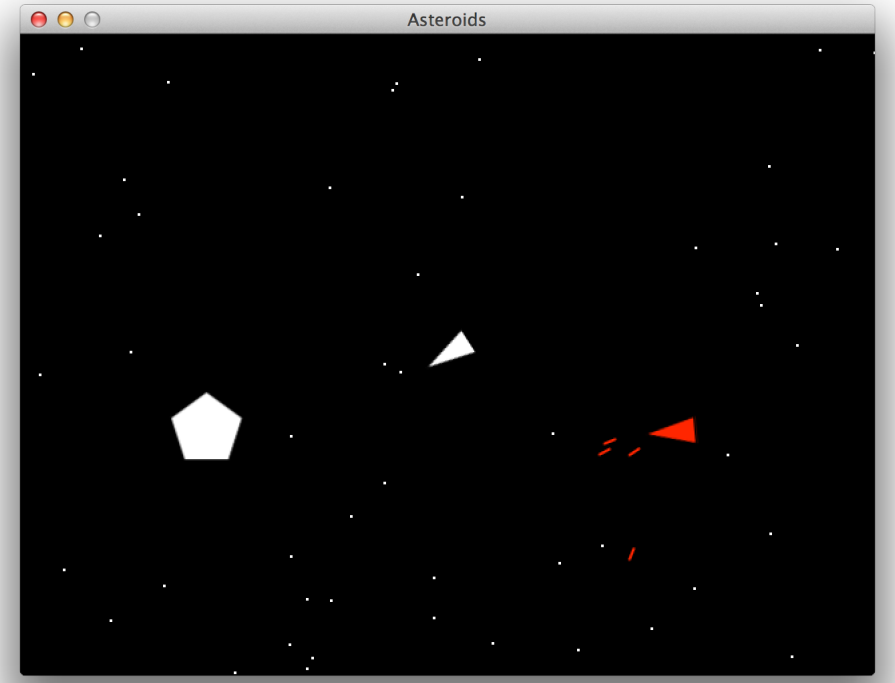
Semester Project:
A Component-based 2D Game with Data-structures, AI and algorithms
SE4-PRO
(10 ECTS)

February 17, 2022



Introduction

In this semester project you will develop a [2D game] with techniques from Component-based Software Engineering, Artificial Intelligence and Data structures and Algorithms. The game will have multiple components that can be added and removed at runtime, and can be implemented and maintained separately.

Requirements

- The game has to included `Player`, `Enemy`, `Weapon`, `GameEngine` and `Map` components.
- The `Player`, `Enemy` and `Weapon` components have to implement provided interfaces that allow the components to be updated and removed dynamically at run-time.
- A component framework has to be applied. The component framework has to support multiple classloaders and component versioning. That is, you can use NetBeans Module System or OSGi.

- At least one component should implement an artificial intelligence technique.
- Data-structures and algorithms have to be applied and documented.
- Finally, a mandatory **report** about the development of the component-based game has to be handed in. The report has to be written in English.

Learning Objectives

- Use component frameworks for development of new applications.
- Use tools to develop component-based software.
- Apply algorithms and data-structures on concrete problems.
- Choose a suitable artificial intelligence technique to solve a particular problem in a project and argue for the choice.
- Work creatively with innovative engineering issues.

Deadlines

FEB 14, 9 AM: Meeting for Presentation of semester project.

FEB 17: Deadline for forming groups.

FEB 24: Deadline for the idea phase (a short project description has to be approved by the project supervisor).

FEB 28, 09:00 AM: Online Meeting for Presentation of the project idea - 5 min presentation per group is expected.

APR 04, 09:00 AM: Online Meeting for Status and demo of the project (Powerpoint + demo).

MAY 16, 09:00 AM: Online Meeting for Demo day for the final project.

MAY 30, 12 PM: Hand-in of the final report.

Report Requirements

Abstract: Define the problem that the paper addresses in context of the game domain and emphases why it is important (motivation). Outline how the developed game addresses the problem – its key characteristics and fundamental principles (establishing a solution). Explain how the developed game improves the situation (the result) and to what extent (evaluating the result).

Introduction: The introduction has to describe the game. Indicate the objectives of your game in relation to the background information and the main issue. The objective of your game should be to address some, if not all of the essential problems, thereby outlining the purpose of the game domain. Describe the high level features of your game.

Requirements: Describe the component-based game in terms of functional and nonfunctional requirements. It is required that components can be added and removed at runtime.

Analysis: Analysis describes only **what** the system should do and not **how** it is done. In analysis, you can come up with a rough draft of the interfaces and the entities of the game. Furthermore, you should document use cases/game-play, the object model using a UML class diagram and the communication between components with sequence diagrams.

Design: The design describes **what** the structure of the system should be to fulfil the requirements. Document the architecture and abstractions of the system. Design develops those abstractions into realizable components. Describe and sketch the component model of the game using a UML component diagram. The component contracts in the system has to be described in terms of pre and post-conditions. Furthermore, the different elements of the game and how they are connected has to be described.

Implementation: In implementation, you document the implementation (code) of the components from design. Describe the details of how the game is implemented. Provide a descriptive explanation of each element in the implementation, and provide arguments for your choices.

Test: Describe how experimental validation was performed through deployment of the game on top of your the component container in a real setting. Test the system's software-abilities such dynamic updates using integration and unit test.

Discussion: Discuss how well the game solved the identified essential problems (dynamic updates etc.). To which extent did your design meet the software-abilities?

Conclusion: First summarize the paper. Remember that you are summarizing the paper for a reader that has read the introduction and the body of the report already, and has a strong sense of key concepts and applied technologies. Explain the potential impacts of your system in relation to the main issue. Direct future work directions related to the main issue. However, this should not be seen as an opportunity to develop new ideas in significant detail, and should be clearly linked to the work described in your paper.

References: Please use IEEE citation style, https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf