# Project 2.0: The second round!

Due date: 11/04/2019
Presentation: 11/04/2019~11/06/2019

## 1 Introduction

In this sprint, you will continue with the development of the *X game* platform. Some (optional) new requirements are provided below.

*Read carefully this document before proceeding!*

## 2 The process

The idea is that you select and work on the highest-priority user stories in your backlog. As soon as you are done with those, you are to perform the necessary scrum ceremonies and start a new sprint. The sprints should not be longer than 30 days, but they could be shorter than that.

Remember to follow the development practices and restrictions briefed in P1.

## 3 Some additional features

Once the set of core features has been implemented in the *X game* platform, you have the option to analyze, design and implement any of the following features or propose a new one (this will require the instructor's approval).

> *"It would be super cool if a user could play against a bot player, you know, something like an AI agent that could decide what's the best move to make??*
>
> *Another nice thing would be the organization of tournaments. So, a user could start a tournament that other users could join. Then the tournament would start with something like eight matches between unique players. The hierarchy of the matches could be randomly determined by the system before the first round of games starts. Badges!! The winner of the tournament would receive a gold badge and the second place would receive a silver badge. Those badges need to become part of the user profile, a public part. The system can use this info in a player ranking table, which should be publicly available.*
>
> *What about a chat or some messaging feature? that would be so cool, too. Is that possible???"*

## 4 Deliverables

There are seven deliverables for this assignment:

1. **User stories and tasks**. They should be uploaded to the GitHub repository. Find a good way of doing so, e.g., in a PDF or MD file.
2. **Kanban board**. A weekly screenshot of the Kanban board should be provided. Find a good way of doing so, e.g., in a PDF or MD file.
3. **Design artifacts**. This document includes the CRC cards and class diagrams. Find a good way of doing so, e.g., in a PDF or MD file.
4. **Source code**. The code should include the JUnit test cases. Well-documented code will be rewarded.

5. **Development manual**. The document should describe how to set up the development environment to work on the project, how to run the system as a developer, and how to run the tests (put yourself in the place of a newcomer—what are the necessary steps for her to start working on the project?).
6. **Traceability link matrix**. This is a spreadsheet document showing the traceability links between user story tasks and the code. While each row represents a task, each column represents an implemented class in the system. A mark in the cell intersecting the task *UC-1A* and the class *MyClassB* indicates that the class *MyClassB* is directly involved in the implementation of the user the task *UC-1A*, as shown next:

| | MyClassA | MyClassB | … | MyClassZ |
|---|---|---|---|---|
| US-1A | X | X | | |
| US-2B | | X | | X |
| … | | | | |
| US-nY | X | | | X |

7. **Output of scrum ceremonies**. Make sure you take notes during the sprint review and the sprint retrospective. Find a good way of sharing your notes, e.g., in a PDF or MD file.
8. **Presentation** (~10 mins). The progress on the project is to be presented during class, on **09/30/2019**. In addition to your progress in the project, you are to present a brief description of the *X game*, as well as any process/product decision you have made. You are also to present the results of your scrum ceremonies.
9. **Peer evaluation** (individual). Each team member is required to fill out the peer evaluation and submit it through Canvas (***do not store it in the team's repository***).

The grades for this assignment will apply to the deliverables uploaded to the GitHub repository before class (i.e., 9:00am).

# 5 Notes

- Delivery dates associated with deliverables will be verified in the repository. Late work policies apply.
- Instead of making assumptions about the requirements, talk with the product owner.
- The analysis, design and implementation of additional features will result in extra points, which are to be added to the general project score.
- Grading criteria:
  - Presentation: 30%
  - Rest of deliverables: 70%
- Points will be deducted if:
  - The submission requirements are not met.
  - You are late with the submission.
- You will not receive credit for this assignment if:
  - You do not submit the deliverables.
  - You do not present during class.