

---

# CBL Game Development

## Assignment 5

*2IP90 Programming*

Q1 2023–2024

---

## 1 Challenge description

You will be making a game using Java and swing in teams of 2 students.

You are free in which game you want to make, as long as you meet the learning goals. During the instruction sessions, you can get feedback on whether your project progress is sufficient.

## 2 Learning goals

The learning goals for this assignment are the following:

- Learn how to apply the concepts learnt during the first weeks in a larger project context.
- Learn how to independently learn 2 different topics relevant to Java programming.
- Learn how to collaborate (part of professional skills).
- Learn how to communicate with group member (part of professional skills).

### 2.1 Topics of choice

During this assignment, you will need to use your basic knowledge of programming to learn about 2 *new or advanced* topics of choice. These must be topics that go further than the material you have been given by the course. You can either pick a new topic, or go more in-depth on topics already covered by the course. Use feedback from the course instructors during the instruction sessions to gauge if your topics are sufficiently new or advanced. The produced code, documentation and project setup must clearly demonstrate that you have understood the material.

Some examples could be: test driven development, advanced object-oriented programming, new Java features, version control (git/hg), user experience (UX), game design, build systems (gradle/maven), continuous integration, algorithms. There are plenty more topics that could be viable, so do not hesitate to show us your ideas for feedback.

## 3 Deliverables

### 3.1 Product backlog

You will create a product backlog for your game before you start programming. You have to hand in this backlog **Tuesday October 10, 2023 before 23:59** on Canvas. This hand-in is required to pass the course.

A product backlog is a list of features (backlog items) that your game will contain. The backlog items need to be relatively small and self-contained, i.e. a small unit you can work on and finish relatively quickly and independently of the other backlog-items. The backlog items need to be sorted by priority from high to low and need to contain the following information per item:

**Name** This should be a succinct descriptor of what your backlog item is about (e.g. “Rendering a chess board”).

**How to demo** How *exactly* are you going to demonstrate that your backlog item is implemented *properly* (e.g. “Start the application, click the *New Game* button and 64 white and black square fields show up in a checkerboard pattern, resize window, board stays square and properly visible”).

**Notes** Here you can elaborate more on the backlog item and write down which learning goals you want to achieve with this backlog item (e.g. “Rendering by painting a JPanel component, dynamically resizing board based on window size — learning how to render on swing components, basic programming.”).

Making the product backlog has 2 main goals:

1. Helping you to make your project ideas concrete.
2. Allowing the instructors to help you scope your project properly.

The exact format or tool you use is up to you — as long as it can be edited easily, has a clear overview of your backlog items, and is ordered/prioritised.

### 3.2 Pitches

You will be pitching the status of your project twice during the instruction sessions. You cannot use slides during your pitch. You may bring a printed version of your product backlog.

**Pitch 1 – Tuesday, October 10** During this pitch you have 90 seconds to pitch your project plan to the teachers. You will give a global overview of your game, and quickly mention the main features your game will have and which learning goals you cover.

**Pitch 2 – Thursday, October 19** During this pitch you have 90 seconds to talk about which features were implemented and what your plans are for the final week. Mention what went well, but also where you encounter difficulties. The instructor can use this information to help you to figure out what to do next.

### 3.3 Constraints

- The game must be written in Java.
- The game must use swing as a UI framework.
- The game must run on Windows, macOS and Linux.
- The game must be playable within 1 minute after starting the program.
- The game must be playable and testable by a single instructor.

### 3.4 Hand-in

Hand in your final product latest **Sunday October 29 23:59 on Canvas**. Your submissions should be a single zip archive containing the runnable project per the constraints in section 3.3. The submission must contain the full source code.

Make sure that the instructor is able to run and assess your project easily and knows where to find what.

## 4 Timeline

See Table 1 for the project timeline. Make sure to also meet outside the scheduled instructions. The average weekly workload should be around *14 hours* per student (including lectures and exam preparation).

Week	Date	Event
5	Thursday October 5, instruction	Creating product backlog
6	Tuesday October 10, instruction	Idea and backlog pitch (1.5 minutes)
	Tuesday October 10, 23:59	<b>Product backlog hand-in</b>
	Thursday October 12, instruction	Standup, work during instruction
7	Tuesday October 17, instruction	Standup, work during instruction
	Thursday October 19, instruction	Progress pitches (1.5 minutes)
8	Tuesday October 24, instruction	Standup, work during instructions
	Thursday October 26, instruction	Standup, work during instructions
	Sunday October 29, 23:59	<b>Project hand-in</b>

Table 1: Assignment Timeline

### 4.1 Standups

During the ‘standups’ in the instruction sessions you briefly give an update on the project. You tell what you have done so far, what you are going to work on and mention whether you are stuck somewhere. This also gives the instructors insight in your project process.

This is a one on one with the instructor and the group members.

## 5 Assessment

The rubric for this challenge can be found in Table 2. Notice that the rubric is vague by design. Therefore, make sure to be present during the instruction sessions and ask for feedback on your project.

Note that it is your responsibility as a student to show that you deserve your grade. So make sure to communicate properly about your achievements. You can use the instruction sessions and pitches to get feedback on the current status of your project.

Rubric	Advanced	Sufficient	Beginner	Insufficient
<b>Functionality</b> <i>Does it run?</i>	(3 pt) Software runs.		(1 pt) Software runs with major glitches.	(0 pt) Software does not run.
<b>Programming basics &amp; code style</b> <i>Variables, control statements, standard library, methods, arrays, code style, ...</i>	(5 pt) Elements are used effectively.	(4 pt) Elements are used, but not efficient and/or effective.	(2 pt) Some elements are missing, wrong or unused.	(0 pt) Not demonstrated, or flat out wrong.
<b>OOP/modularity</b> <i>Classes, interfaces, inheritance, polymorphism, ...</i>	(5 pt) Used properly and shows insight.	(4 pt) Used properly.	(2 pt) Used, but some concepts are wrong and/or missing.	(0 pt) Not demonstrated, or flat out wrong.
<b>GUI</b> <i>Swing</i>	(4 pt) Elements used properly.		(2 pt) Elements missing and/or wrong.	(0 pt) Not demonstrated, or barely.
<b>Topic of choice 1</b> <i>See 2.1.</i>	(4 pt) Shows insight.	(3 pt) Done.	(1 pt) Done, but with some problems.	(0 pt) (Almost) absent, or wrong.
<b>Topic of choice 2</b> <i>See 2.1.</i>	(4 pt) Shows insight.	(3 pt) Done.	(1 pt) Done, but with some problems.	(0 pt) (Almost) absent, or wrong.
<b>Total</b>	(25 pt)			

Table 2: Rubric CBL Game Development