HOGESCHOOL ROTTERDAM / CMI

Project 4

Developing a cross platform application

INFPRJ00-4

ECTS: 4
Course responsibles: M. Abbadi, L. Muilwijk

Description of the course

Cursusnaam:	Project 4 – Developing a cross platform application
Curuscode:	**INFPRJ00-4**
Aantal studiepunten en studiebelastinguren:	This course provides you with four (4) study points, which corresponds to a workload of 112 hours. The recommended distribution of these 112 hours during the study weeks is as follows: <u>Supervised lectures:</u> Kick-off: 3 * 50 minutes 2,5 hours Project lesson (for 3 weeks): 6 * 50 minutes 15 hours Presentation of the product: 3 * 50 minutes 2,5 hours <u>Unsupervised hours:</u> Time to work on the project incl. literature study 92 hours **Total** **112 hours**
Vereiste voorkennis:	Basic programming and database knowledge is required.
Werkvorm:	Project-based education (group-work)
Toetsing:	Examination is based on the delivered product and the process of the project.
Leermiddelen:	Development tools, Dev4 literature
Draagt bij aan competentie:	▪ Advising (**A**dviseren) ▪ Design (**O**ntwerpen) ▪ Implementation (**R**ealiseren)
Leerdoelen:	▪ [**O1**] You can represent the application and the design patterns used in it through UML ▪ [**S1**] You can clearly and convincingly present your product to an audience with different (technical) backgrounds ▪ [**R1**] You can develop a mobile application ▪ [**R2**] You can adapt a mobile application to different platforms with design patterns
Inhoud:	You learn to work in a team context (*process*) and to realize a project assignment (*product*) for a client.
Opmerkingen:	Presence is **required**. Students make teams themselves. They have to communicate these to the tutor before the end of the first day of the project (through e-mail). Teams should be 4 or 5 students each. The tutor has to give approval and has the ability to make some adjustments if needed. *Note*: Teams that, during the course of the project, lose team members and remain with 3 (or less) students, will discuss with their project teachers about adjusted criteria for the evaluation.
Cursusbeheerder:	M. Abbadi, L. Muilwijk
Datum:	9 June 2017

1. General information

1.1 Introduction

You have already taken part in 3 projects. You have gained substantial experience in working together in a project team and you have mastered the Scrum method. You might have been a Scrum Master once and you have had several opportunities to develop products using a new programming language. You delivered several (programmed) shippable products and have grown more competent in the working field of a software engineer. This project will focus on a new aspect of this field: writing a mobile application and trying to abstract it to a cross-platform application by means of design patterns.

You will work in a project team again, in a 3 week time frame. Once again you will be working together on a case that has to be finished in the end of week 10. Your Product Owner and Tutor will set the rules with regard to delivery and will guide you through the process. Feedback will be given regularly.

1.2 Relationship with other courses

You will need to apply all the knowledge learned during the three courses (Skl, Anl, Dev) in the project itself.

The knowledge acquired during the Skills course (INFSKL02-1) is tightly connected and applicable to every project you will encounter. The courses of Analysis INFANL02-3 and Development INFDEV02-4 are also crucial to bring this project to a satisfying end. In the ANL course you learned how to schematically represent applications with UML. In the DEV course the focus was on design patterns.

1.3 Learning materials

Mandatory:
- Google Classroom or N@tschool (for deliveries)
- Scrumboard (Trello, et al.)
- Github, https://github.com
- DEV, ANL, SKL literature

Facultative:
- MonoGame (C#): http://www.monogame.net
- LibGDX (Java): https://libgdx.badlogicgames.com
- Franken, M. (2013), *Scrum voor Dummies* (1e druk), Amsterdam: Pearson Education Benelux B.V., ISBN 978-90-430-2403-7
- https://www.scrumalliance.org/
- Coursera: https://www.coursera.org (online courses)
- Udacity: https://www.udacity.com (online courses)
- Wireframes: http://ninjamock.com
- Q&A for technical help: http://stackoverflow.com
- Android Developers: http://developer.android.com/index.html
- Publishing checklist https://developer.android.com/develop/quality-guidelines/core-app-quality.html

2. Program and contents

2.1 Assignment

During Project 4 you will create a cross-platform application. This application should at least run on desktop computers and on a mobile platform. The focus in this project is on an architecture that should be able to reuse most of the shared logic code. This can be achieved through the use of design patterns.

- The projects are located in Attachment 5. You can choose one of these projects or create and pitch one to your P.O. in week 8
- You must create a sprint backlog approved by your P.O.
- In case your application contains visualizations: legend, title and values on the axes should be indicated accurately

2.2 Week scheme

The project covers the last three weeks of OP4: week 8 (sprint 1), week 9 (sprint 2) and week 10 (sprint 3). In this project it is *again* mandatory to use Scrum as a project method (see Attachment 1 for more details).

In the following table you can see the lessons of each week and the corresponding deliveries. Deliveries must be done **no later than 12 hours** before the start of the corresponding lesson. For example, if your lesson is at 10.30 of Wednesday, then you must deliver before 22.30 of Tuesday. The deliveries will be done through Google Classroom or N@tschool[1].

Week	Day (see schedule for detail)	Teachers present[2]	Topic	Deliveries (deadline: 12 hours before the lesson!)
8	Monday	P.O. and Tutor	Kickoff together with all the first-year students	
8	~half of week	Tutor	Assessment on use of Scrum and explanation on the Tutor part	
8	~end of week	P.O.	Review sprint 1	-- UML class diagrams (draft 1) representing the entire application emphasizing the design patterns used; -- Product Backlog and Sprint Backlog (for sprint 2); -- Shippable product [no upload needed; you must be able to show it during the review];
9	~half of week	Tutor	Assessment on use of Scrum and team functioning	

[1] The scrum master of each team will have the duty of uploading the documents to deliver for his/her team in a compressed file called “INF1X – Group Y – Week Z”.
[2] If compatible with personal work schedules, Tutors are invited to attend the P.O.s lessons, and vice versa.

9	~end of week	P.O.	Review sprint 2	-- UML class diagrams (draft 2) representing the entire application emphasizing the design patterns used; -- Product Backlog and Sprint Backlog (for sprint 3); -- Shippable product [no upload needed; you must be able to show it during the review];
10	~start of week	P.O.	Assessment on progress (*on request*)	
10	~half of week	Tutor	Assessment on use of Scrum and team functioning	
10	~end of week	P.O. and Tutor	Final presentation, including demonstration and evaluation (review sprint 3)	-- Complete documentation (UML class diagrams with explanations); -- Final version of code with comments; -- Screenshots of the application; -- Presentation slides (including short video)

Important note: for all deliveries, read the evaluation attachments for further details.

3. Evaluation

	Evaluated by	Evaluated through	Partial result
Collaboration (team) and project management (individual)	Tutor	*Attachment 1 (Evaluation form Tutor)*	Yellow cards
Intermediate deliveries	P.O.	*Attachment 2 (Evaluation form PO)*	Yellow cards Shippable product
Individual code contribution	P.O.	*Attachment 3 (Evaluation form PO)*	Yellow cards
Final products	P.O. & Tutor	*Attachment 4 (Evaluation form Final Products)*	Yellow cards Shippable product

- The final grade is the sum of the partial results, with a maximum of **10 points** and a minimum of **1**
- The total amount of yellow cards could cost you a maximum of **3 points**

3.1 – Examples

The following partial grades:
- 2 yellow cards from Attachment 1
- 1 yellow card from Attachment 2
- No yellow cards from Attachment 3 (individual)
- 6,5 points from Attachment 4

bring to a final grade of (-1) + (-0.5) + 6.5 = **5** (onvoldoende) → resit

The following partial grades:
- 2 yellow cards from Attachment 1
- 1 yellow card from Attachment 2
- No yellow cards from Attachment 3 (individual)
- 7 points from Attachment 4

bring to a final grade of (-1) + (-0.5) + 7 = **5.5** (voldoende)

The following partial grades:
- **Individual No Go** from Attachment 1
- No yellow cards from Attachment 2
- 1 yellow card from Attachment 3 (individual)
- 6 points from Attachment 4

bring to a final grade of onvoldoende → herkansing

3.2 – Resit

In case of an insufficient grade (*onvoldoende*) for INFPRJ00-4, the following scheme applies:
- If you got an individual No Go, then you **cannot** repeat the project this year, but in OP1 of next school year;
- If your team did not complete the “basic features” (see Attachment 3), then you **cannot** repeat the project this year, but in OP1 of next school year;
- In all other cases, you have to individually select at least one *additional feature* (chosen by the P.O. from a list of features specific for the resit) by the end of **week 11** and submit it before the last week of August.

If you do not succeed at the resit (or if you cannot repeat the project this year), you will need to follow this course again during next school year.

Attachment 1 – Evaluation form [Tutor]

CLASS: INF1…, Group…

The Tutor part will be evaluated in two parts; an individual grade (Scrum) and a team grade (team functioning). Attendance is obligatory to get a grade for the project. For the individual grade and the team grade you will get yellow cards, with a combined maximum of 3 yellow cards. Meaning a maximum of -1.5 points.

Individual grading [learning goal: S2]

Attendance will be checked during every Sprint. Only when a valid reason is given to the Tutor prior to the lesson, an exception might be granted only once. When you have not met the attendance standards you will receive a No Go (meaning you have not taken part in the course). A No Go can also be given if a student has delivered considerably less work during the entire project or almost all weeks. He or she has been a burden to the group's development and cannot really show the amount of work he or she has done individually. The Tutor will write down the attendance (and possible No Go's) in the following table.

Student's name:	Week 8	Week 9	Week 10	Total sum of week 9 and 10 (Go or No Go)
	Go / No Go	Go / No Go	Go / No Go	Go / No Go
	Go / No Go	Go / No Go	Go / No Go	Go / No Go
	Go / No Go	Go / No Go	Go / No Go	Go / No Go
	Go / No Go	Go / No Go	Go / No Go	Go / No Go
	Go / No Go	Go / No Go	Go / No Go	Go / No Go

During every Sprint Scrum has to be applied. When the Tutor meeting takes place, your Tutor will ask your team to provide evidence for implementing the Scrum method (see the row below for the Scrum parts). Every student should be able to provide evidence for applying the Scrum method effectively. If Scrum is not implemented every week (or every Sprint) the individual student will get a yellow card.

Student's name:	Week 8 Scrum assessment	Week 9 Scrum assessment	Week 10 Scrum assessment	**Total yellow cards**

Scrum parts:

- Sprint Backlog (as a team member you have to make sure that user stories, time estimate, and acceptance criteria are well defined. You are also aware what tasks will be done during this Sprint)
- Retrospective (as a team member you were present during every retrospective and you reflect clearly on working together, this is shown in the visible product)
- Burndown chart (as a team member you are responsible that a chart is provided every Sprint and it is clearly visible how many user stories have been finished during a Sprint)
- Scrumboard (as a team member you have to maintain the board and have to make sure that tasks are roughly evenly distributed between the members)

Team grading [learning goal: S1]

Your Tutor will ask you every Sprint about your team functioning and team membership and will determine whether you have been an effective team member. The P.O. will try to establish if you deserve to get the team grade and he will take a look at your individual contribution to the code.

When you are a team, you should actively work together. This means that every team member delivers a substantial contribution and helps and motivates members to deliver a good contribution too. To make sure that the process runs smoothly, you stick to the rules in the cooperation contract and feel responsible for the process of the project. When the Tutor meeting takes place, your Tutor will ask you to inform him/her about team functioning (for example Retrospective, relevant anecdotes, notes etcetera). It is your job to prepare this before meeting your Tutor. Through an assessment, the following criteria will be evaluated. In week 8 you will get feedback on your team spirit.

You are not allowed to miss any boxes in week 9 and 10, all team functioning criteria have to be met. If not, the team will receive a yellow card. In total, you can get 3 yellow cards. Every yellow card, means - 0.5 points. You can get a maximum of - 1.5 points.

Team functioning	**Week 9**	**Week 10**	**Total yellow cards**
Team members help each other when necessary			
Everybody sticks to the rules defined in the cooperation contract			
Team members give constructive feedback to each other and about each other			
The team members are capable of reflecting on the process critically			
The tasks are evenly distributed (if not, there is a clear explanation provided)			

Attachment 2 - Evaluation form (Intermediate deliveries) [P.O.]

CLASS: INF1…., GROUP …

Sprint 1 review

CRITERIA	POINTS
• The draft of the UML class diagram (version 1) is delivered on time, representing the entire application emphasizing the design patterns that you plan to use • There is a shippable product shown during the review	YES => 0 points NO => Yellow card [3]
• There is a Product Backlog (PB) • The PB contains enough items to complete the product • Each item of the PB is associated to a unique ID	YES => 0.5 point NO => 0 points
• The sprint 2 backlog proposal is realistic (considering the workload) • The sprint 2 backlog contains all needed information o Selection of items from the PB and for each one: ID, user story, Moscow priority, planning poker, tasks, acceptance criteria	YES => 0.5 points NO => 0 points

Notes on feedback given:

_____

_____

_____

_____

_____

_____

[3] See *Attachment 3* for consequences of yellow cards on the grade.

Sprint 2 review

CRITERIA	POINTS
• The improved draft of the UML class diagram (version 2) is delivered on time, representing the entire application emphasizing the design patterns that you plan to use o The feedback received in week 8 has been applied and is visible in the improved UML draft • There is a shippable product shown during the review o The feedback received in week 8 has been applied and the results are visible in the application	YES => 0 points NO => Yellow card
• The progress of the sprint backlog items planned for this sprint is...	Good => 1 point (you delivered all items) Sufficient => 0.5 points (you missed at <u>most</u> one item) Insufficient => 0 points (otherwise)
• The sprint 3 backlog proposal is realistic (considering the workload) • The sprint 3 backlog contains all needed information o Selection of items from the PB and for each one: ID, user story, Moscow priority, planning poker, tasks, acceptance criteria	YES => 0.5 points NO => 0 points

Notes on feedback given:

_____

_____

_____

_____

_____

_____

Attachment 3 - P.O

During every Sprint each student is asked to present/discuss their work within the implementation of the application. The distribution of the work must be fair. This means that one student may not perform considerably more work than another. If a student is incapable of providing/explaining their work, a yellow card is given.

For the individual grade and the team grade you will get yellow cards, with a combined maximum of 3 yellow cards. Meaning a maximum of -1.5 points.

Student's name:	Week 8 Work assessment	Week 9 Work assessment	Week 10 Work assessment	**Yellow card**

Attachment 4 – Evaluation form Final Product [PO & Tutor]

During the last review, each team presents its product. During the presentation you must show the *features* of your product clearly, so that the Product Owner can evaluate the final result. The quality of the presentation will be evaluated by the Tutor. The final grade for this part is obtained by summing up the points for each criteria.

TEAM POINTS

CLASS: INF1…., GROUP …

CRITERIA	POINTS
• The complete product is delivered on time	YES => 0 points NO => Yellow card
• The complete product contains: code, improved UML class diagrams with explanation and screenshots of the application • Code is commented • Use an object oriented language (Java or C#)	**Prerequisite** to get the points of Week 10
The application contains the following ***basic features/attributes***: • Cross-platform • Stable (does not crash) • All approved Must-have features [M] are implemented • There is evidence of the implementation of at least *three distinct* design patterns	YES => 3.5 points NO => 0 points
The application includes the following ***additional features****:* • The same logic code is reused for a *desktop application* through a different visualization layer (for instance, you did not use Monogame for both implementations, but Monogame for mobile and WPF for desktop) [2.0p] • All *Should-have* (S) *features* (approved by the P.O.) have been implemented [1.0p] • There is *one instance* of a database used among all implementations (for example, a database could be stored on the cloud and be used by both desktop and mobile versions) [1.0p]	**NOTE**: If you do not get the points for the *basic features* or the teachers are not satisfied enough by their quality, no points for *additional features* will be given. Score of additional features: _____
The final **presentation** has the following features: ● Each member of the group contributes evenly to the presentation ● The presentation covers all relevant aspects (max. 15 min) ● The presentation is understandable for everybody (group members, fellow students, P.O. and Tutor) ● The presentation is well structured	YES => 0 points NO => Yellow card
Total team points:	...

FINAL GRADE

CLASS: INF1…., GROUP …

The individual student or the team received yellow cards (with a maximum of 3 yellow cards) [P.O. part]	0 yellow cards => 0 points 1 yellow cards => -0.5 points 2 yellow cards => -1.0 point 3 yellow cards => -1.5 points
The individual student or the team received yellow cards (with a maximum of 3 yellow cards) [Tutor part]	0 yellow cards => 0 points 1 yellow cards => -0.5 points 2 yellow cards => -1.0 point 3 yellow cards => -1.5 points

Student's name:	Total team points:	Individual yellow cards [Tutor]	Team yellow cards [Tutor]	Individual yellow cards [P.O.]	Team yellow cards [P.O.]	Final grade:

Attachment 5 – Case examples

In this attachment you will find several example cases for Project 4. It is recommended - but not mandatory - to select from these cases. If you should decide not to use any of the provided cases, you must design your own case and have it approved by your P.O.

Open Day
Hogeschool Rotterdam is interested in an application for their open day. Students should be able to find all information relating to the open day and the available educations. Furthermore, the application should provide a means of asking questions to which teachers can respond.

The application should support the following features:
- Map with schedule and information about the activities in each space
- Means of asking and answering questions through the internet
- Information screens for all offered educations
- Information about other locations of the Hogeschool

Portable Recipes
This application is really a digital cookbook. It offers a collection of recipes and a suggestion algorithm based on recipes marked as favored.

The application should support the following features:
- Offer a collection of recipes
- Support the ability to rate and bookmark recipes
- An algorithm that suggests other recipes based on the user's ratings
- A gallery of recipes that can be filtered and searched by the user

Robot Party
This is a reimagination of the classic arcade hit *Robotron: 2048*. It is a real-time action game involving laser projectiles, hostile robots and helpless victims to be rescued.

The application should support the following features:
- Several game stages
- Multiple characters with different behaviours
- Movable player character
- Firing of projectiles that collide with characters and yield points
- Should reasonably reflect the original game experience

Your own idea
…