

Moduleteam: *Dirk Vandycke - Thijs Martens - Kristien Roels - Corneel Theben Tervile - Jonas Maes – Jill VandenDriessche - Bernard Samyn*

# **Dominion 2016**

## **Programmeerproject Projecten I**

# 1 Situering van het project

Dit project kadert in de module Projecten I (9 ECTS punten)

Deze bestaat uit de partims:

- Projectmanagement (2 ECTS punten)
- Projectcommunicatie (3 ECTS punten)
- Programmeerproject (4 ECTS punten)

Zie ook de studiegids voor meer informatie

## 2 Beschrijving van de opdracht

### Dominion

Dominion is wat men noemt een 'deck building (card) game' en welbekend onder gamers (de niet-digitale welteverstaan). Bedoeling van dit programmeerproject is een server met webclient implementatie ervan neer te zetten.

De te behalen competenties voor de andere partims van deze module (projectmanagement en projectcommunicatie) worden geëvalueerd in dit project.

Er worden aldus 4 zaken verwacht van jullie:

1. Een **analyse van de software**
2. Een **werkende** versie van de **game** in **Java/HTML5/CSS/JavaScript/mysql**
3. Een zelf opgesteld **projectmanagementdossier**
4. Een verzorgde technische **eindpresentatie**

## 2.1 Analyse van de software

### 2.1.1 Digitaal dossier - analyse

Er wordt een (digitaal) dossier aangelegd voor de ontwikkeling en analyse van deze game.

De gebruikte analysemethodiek(en) is/zijn vrij naar keuze, enkel **aan onderstaande minima moet worden voldaan**:

#### 2.1.1.1 Informatieanalyse

Het is de bedoeling dat een databank wordt opgesteld waarbij de informatie die in de casus beschreven wordt vervat zit.

##### 1) Opstellen van een conceptueel schema in ERD

Op basis van de beschreven opdracht stel je een conceptueel model op met behulp van ERD. Als je een syntax van ERD gebruikt die behoorlijk wat plaats in beslag neemt gebruik je per entiteit een aparte pagina. Je kan dan helemaal als laatste een ERD gebruiken met enkel de entiteitstypes en relatietypes.

##### 2) Mappen van het conceptueel schema naar het relationeel datamodel

Map het conceptueel model naar het relationeel model. Gebruik een duidelijke schematechniek die de tabellen en relaties ertussen weergeeft.

#### 2.1.1.2 Databank

Het relationeel model met de nodige constraints wordt geïmplementeerd in MySQL.

#### 2.1.1.3 Wireframes

De volledige applicatie wordt uitgewerkt in high fidelity wireframes. Er mag geen enkele twijfel bestaan over wat er gebeurt “in geval van”. Dit is duidelijk in de wireframes.

(Tip: Indien je maar 3 schermen hebt in je wireframes ben je niet goed bezig...)

Deze documenten dienen als afspraak tussen jullie teamgenoten alsook de klant. Zie dit als onlosmakelijk onderdeel van het contract.

Je bent vrij van keuze voor een wireframing tool (potlood, pen, software voor clickable mockup), maar het moet vooral **LEESBAAR** zijn.

Minimale vereisten wireframes:

- **Elk scherm** is omschreven
- **Elk verband tussen de schermen** is duidelijk, hetzij op dezelfde documenten, hetzij door middel van een sitemap, hetzij door een clickable mockup. Dit wil zeggen dat alle knoppen duidelijk naar ergens leiden en de gebruiker nergens “vast” mag komen te zitten.  
Een scherm kan niet bestaan zonder verband, anders is deze niet toegankelijk / verlaatbaar voor de gebruiker

## 2.1.2 Basisverwachtingen

### **1) *Gameplay concept multi-player single screen:***

Twee spelers moeten op hetzelfde scherm tegen elkaar kunnen spelen. Dit betekent dat op 1 machine de controle over het spel moet worden afgewisseld onder de spelers op aangeven van de software. Denk goed na over hoe je zoiets in een GUI implementeert.

Alle basisregels in de handleiding moeten geïmplementeerd worden en waar mogelijk door de software worden afgedwongen. Een Nederlandstalige versie zowel als een Engelstalige versie van de spelregels van het basisspel zijn te vinden op de cursus-site. Hierbij moet de historiek van alle gespeelde games worden bijgehouden zodat het (in principe) mogelijk is het spel terug af te spelen.

Het terug afspelen van een eerdere game moet niet worden geïmplementeerd in de UI maar wel in het databank en objectmodel aanwezig zijn. Het moet wél mogelijk zijn een game te save en later terug verder te spelen.

De **GUI** wordt opgesteld in **HTML5/CSS** en waar nodig **JavaScript**. Grafische elementen dien je niet zelf te maken maar mag je gebruiken van bestaande materialen. Denk wel aan bronvermelding!

Het is niet de bedoeling (lees: verboden) dat het model aan de client-zijde wordt geïmpelmenteerd in JavaScript maar wel aan de serverzijde in Java. Javascript dient dus enkel om informatie van/naar de server op te halen/te sturen en waar nodig animatie en event handling in de GUI.

Om te bewijzen dat dit zo is verwachten we ook een desktop applicatie in Java Swing (dus JFrames e.d.) waarmee het spel kan worden gespeeld op een gelijkaardige manier. De desktop applicatie en de webclient staan los van elkaar maar **moeten** dezelfde modelklassen en databank gebruiken.

Een test daar is het save van een game in de webclient en het verder spelen ervan in de Java GUI.

**De Java GUI mag zeer beperkt zijn** en dient enkel om te bewijzen dat het klassenmodel en databank los kunnen staan van de webclient.

Unit test voor het ganse Java model zijn verplicht. Aan te raden is dus deze mee te laten groeien met de code (al dan niet via TDD). We verwachten minstens de implementatie met drie spelers. Meer mag.

## **2) *Gameplay concept multi-player multi-screen (optioneel):***

Als **uitdaging** mag geprobeerd worden om een multiscreen versie te maken. Dit is echter ondergeschikt aan de single-screen versie die we zeker afgewerkt verwachten.

### **2.1.3 Wat dien je in? (deliverable 1)**

Dien **één pdf-document in per groep op de opdrachten ruimte in Leho van ‘Projecten I’**.

Naamgeving: analysedossier-groepX.pdf

Vermeld op de eerste pagina duidelijk de namen van de teamleden.

Het document bevat **DUIDELIJK LEESBAAR** het ERD diagramma, het relationeel model en de wireframes - of een link ernaar indien clickable mockup. Uiteraard hou je rekening met alle vereisten die hierboven vermeld staan.

#### **3) Verdere uitbreidingen**

Ook uitbreidingen zijn niet verplicht maar worden wel beloond. **In elk geval zijn deze ondergeschikt aan de vereiste voor een werkende single-screen versie.** Mogelijke uitbreidingen zijn:

- speleridentificatie en opslag van games/scores/etc ...
- computerspeler(s)
- meer dan drie spelers
- uitbreidingssets van Dominion (er zijn er verschillende)
- deck builders (elke game start met een aantal verschillende kaarten - zie spelregels)
- JSON export van een opgeslagen sessie, al dan niet via HTTP aanspreekbaar

## 2.2 Werkende (beperkte) versie van de game in HTML, CSS en JavaScript, alsook in Java

### **1) Praktische uitwerking**

Je maakt een 2 werkende versies van de game volgens bovenvermelde basisvereisten in zowel HTML/CSS/JS alsook in Java, mits minimale GUI vereisten voor deze laatste zoals eerder vermeld.

Je werkt hierbij in teamverband volgens de aangeleerde principes van het objectgeoriënteerd programmeren alsook werkmethodeken (Kanban, scrum, ...).

Zorg dat je applicatie schaalbaar is waar gevraagd. Probeer aandacht te hebben voor het herkennen en toepassen van de principes achter design patterns en het refactoren in die richting.

### **2) Databank**

Je gebruikt een MySQL databank om de nodige gegevens in op te slaan. Kijk goed in de basisvereisten wat er minimum verwacht wordt bijgehouden te worden.

Gebruik je ER-diagramma uit de analyse ter ondersteuning.

Voor het onderhouden van de database wordt **geen CMS** uitgewerkt, aanpassingen van de data mogen rechtstreeks in de databank zelf gebeuren tijdens het ontwikkelen.

### **3) Unit testing**

De applicatie bevat de nodige unit tests. Het gaat hier vooral om de controle van het correct functioneren van de klassen. Een game simulator is niet nodig. UI klassen hoeven dus niet getest te worden. Ze mogen evenwel geen code bevatten die in het model thuishoren. Dit behelst ook integratietesten (om de goede samenwerking tussen klassen aan te tonen/te bewaken).

### **4) Versioning**

Voor versioning gebruik je git.

Maak een account aan op github en geef ook toegang aan de lectoren om je repository te bekijken. Een intro op het gebruik van github, jsp/servlets en jdbc (voor databanktoegang) wordt de maandagmorgen van de eerste projectweek gegeven.

### 2.2.1 Wat dien je in? **(deliverable 2)**

Dien **één zip file in per groep op de opdrachten ruimte in Leho van 'Projecten I'**.

Naamgeving: dominionsource-groepX.zip

Structuur:

- java-backend/
- java-frontend/
- html-frontend/

Zoals je zit is het project mooi opgesplitst in 3 lagen. Werk van het begin zo om voor jezelf duidelijk het onderscheid te maken.

Uiteraard hou je rekening met alle vereisten die hierboven vermeld staan.

## 2.3 Projectmanagementdossier

**Deze opdracht maak je tijdens de uitwerking van het Project, maar valt onder het partim "Projectmanagement" en wordt gequoteerd onder dit partim als permanente evaluatie.**

Jullie werken tijdens het project met MS Project waarin we de rollen in een bedrijf zo goed mogelijk zullen simuleren.

**Wat moet er zeker aanwezig zijn:**

### 2.3.1 Toegewezen Takenlijst met deadlines

Stel een takenlijst op (dit is uiteraard een dynamisch gegeven).

Groepeer je taken volgens fase:

Bvb. Analyse, ontwikkeling, testing, ...



**Wijs elke taak toe aan een teamlid.** Stel deadlines in per taak.

Stel een logische koppeling in tussen de taken.

Werk in je planning ook zogenaamde “**werksessies**” in. Dit zijn momenten waarbij je samenkomt om het analyseren / programmeren / ... door te nemen of uit te voeren.

## 2.3.2 Milestones

Voeg een aantal belangrijke **milestones** toe. Controlemomenten waar je naartoe werkt.

Bvb. Analysebespreking, bètaversie klaar, presentatie, ...

## 2.3.3. GANTT-diagram

Zorg voor een overzichtelijk **GANTT-diagram** met een overzichtelijke weergave in de tijd van de uit te voeren taken, waarop ook de **milestones** zijn aangeduid.

## 2.3.4. Planning vs. eigenlijk gespendeerde tijd

Zorg dat je bij de opstart van het project het initieel plan instelt als **basislijn**. Pas tijdens het project de timing van elke taak correct aan.

Zorg dat je op het einde van het project een analyse kan maken **in MS Project** waarbij je de vergelijking maakt tussen de oorspronkelijk ingeschatte timing (basislijn) en de werkelijk gespendeerde tijd.

## 2.3.5. Loonkostenberekening

Beschouw 2 teamleden als ‘junior’ met een kostprijs €50 per uur, en beschouw 2 teamleden als ‘senior’ met een kostprijs van €80 per uur (indien niet met 4 in een groep, zorg voor een evenwichtige verdeling).

Bereken de geschatte kostprijs van het project op basis van de oorspronkelijke planning en vervolgens ook de uiteindelijke kostprijs van het project op basis van de uiteindelijk gespendeerde tijd.

### 2.3.6 Wat dien je in? (deliverable 3)

Dien **één pdf-document in per groep**.

Naamgeving: projectmanagementdossier-groepX.pdf

Vermeld op de eerste pagina duidelijk de namen van de teamleden.

Het document bevat **DUIDELIJK LEESBAAR** de takenlijst, het GANTT-diagram, de analyse van de timing en de loonkosten (met alle vereisten die hierboven vermeld staan).

## 2.4 Eindpresentatie

Deze presentatie is voor een technisch publiek, doch vergt uiteraard enige setting. Voorzie een werkende demo van je applicatie. Indien je bepaalde functionaliteiten wenst te tonen die niet makkelijk in een korte tijdspanne te demonstreren zijn (bvb. Game over, herspelen vorig spel, ... ), voorzie dan extra demo's speciaal aangepast voor hetgeen je wenst te tonen. Hierbij kan je indien nodig een en ander hardcoderen "for demonstration purposes", of bvb. initialiseren van een andere dataset.

Wees je ervan bewust dat we elk teamlid individueel zullen ondervragen over het technische luik van dit project. Elk stuk code dient door elk teamlid gekend te zijn! Uiteraard verdelen jullie het werk onder elkaar, maar iedereen dient kennis te hebben van de werking van de volledige codebase. Dit dient ter illustratie van "real life" situaties, waarbij ook vaak iemand uitvalt (ziekte, ander werk, ...) en anderen aan de codebase moeten verder werken.

### 2.4.1 Wat dien je in? (deliverable 4)

Dien **één pdf-document/ppt-document in per groep op de opdrachten ruimte in Leho van 'Projecten I'**.

Naamgeving: presentatiedominion-groepX.pdf

# 3 Doelstellingen van het project

Naast de competenties vermeld in de studiefiche zijn de doelstellingen van dit project verder drieledig:

## **Doelstelling 1: Het kunnen uitvoeren van een informatiemodellering**

Dit bevat:

- Een informatieanalyse en -modellering kunnen uitvoeren, waarbij de bereikte resultaten geëvalueerd dienen te worden. Het bekomen resultaat kunnen implementeren in een relationele databank.

## **Doelstelling 2: Het kunnen programmeren volgens de principes van een gelaagd OO model in teamverband**

Dit bevat:

- In teamverband een gelaagde objectgeoriënteerde applicatie met een professionele grafische gebruikersinterface kunnen ontwikkelen. Deze applicatie maakt gebruik van een relationele databank om de gegevens op lange termijn te bewaren.
- De objectgeoriënteerde principes en de principes van gelaagd programmeren kunnen toepassen. Het herkennen en toepassen van de principes achter design patterns en het refactoren in die richting.
- Het ondersteunen van de beheersbaarheid van een SO project met unit- en integratietesten.
- Het correct werken met een versiecontrolesysteem zoals git om het gemaakte werk onderling te kunnen onderhouden

## **Doelstelling 3: Het projectmatig aanpakken van een uitvoerbaar werk binnen een gegeven kader**

Dit bevat:

- Het opstellen van een eigen ontwikkelingstraject met eigen planning en tussentijdse milestones die de werkelijkheid weerspiegelen. De planning kadert in een raamwerk van opgelegde deadlines waaraan moet voldaan worden
- Het onderling toekennen van bepaalde rollen aan de verschillende teamleden en correct en tijdig uitvoeren van de toegewezen taken

# 4 De begeleiding van het project

## 4.1 Begeleiders

- Jonas Maes [Jonas.Maes@howest.be](mailto:Jonas.Maes@howest.be)
- Jill VandenDriessche [Jill.Vandendriessche@howest.be](mailto:Jill.Vandendriessche@howest.be)
- Dirk Vandycke [Dirk.Vandycke@howest.be](mailto:Dirk.Vandycke@howest.be)
- Kristien Roels [Kristien.Roels@howest.be](mailto:Kristien.Roels@howest.be)
- Corneel Theben Tervile [Corneel.Theben.Tervile@howest.be](mailto:Corneel.Theben.Tervile@howest.be)
- Thijs Martens [Thijs.Martens@howest.be](mailto:Thijs.Martens@howest.be)
- Heidi Terryn [Heidi.Terryn@howest.be](mailto:Heidi.Terryn@howest.be)
- Bernard Samyn [Bernard.Samyn@howest.be](mailto:Bernard.Samyn@howest.be)

## 4.2 Organisatie begeleiding

Er zullen **coachingmomenten** worden voorzien waar een begeleider aanwezig is om advies te geven en het reeds gedane werk te evalueren en bij te sturen. Iedereen is steeds op die momenten aanwezig tenzij anders aangegeven. Voor alle administratie wordt de cursussite op Leho gebruikt.

# 5 Afspraken, coachingmomenten en deadlines

## 5.1 Coachingmomenten en deadlines

### Vrijdag 18 maart

*Wat:* Plenaire toelichting project en finale groepsvorming (in het begin van de les Algoritmen en Datastructuren)

*Verplicht:* Ja

### Projectweek 1

*Verplicht:* Ja

## Projectweek 2

*Verplicht:* Nee

### Vrijdag 20 mei

*Wat:* Indienen eindresultaat; peer assessment en sourcecode (als zip op Leho dropbox) EN op github

*Verplicht:* Ja

*Wat indienen:*

- ALLE source files van je project
- Digitaal projectdossier volledig up-to-date
- Projectmanagementdossier volledig up-to-date
- .jar executable van de applicatie
- Peer assessment (Leho)

### Donderdag/vrijdag 26/27 mei

*Wat:* Eindpresentatie

*Verplicht:* Ja

Iedereen wordt verwacht op alle presentaties. Jullie zullen elkaars werk zien en ook de feedback van de docenten horen.

**VOOR** de presentatie wordt door elk teamlid een peer assessment ingevuld waarbij hij zijn teamleden naar prestatie een score toekent. Deze scores worden niet gepubliceerd en zijn strikt vertrouwelijk.

Opgelet: Lokalen per sessie en eventuele wijzigingen in de planning worden via Leho gecommuniceerd.

## 5.2 Indienen

Zie 4 deliverables aangeduid (in rood) doorheen het document. Gevraagde onderdelen zijn per deadline afhankelijk. Deadlines worden apart gecommuniceerd via Leho.

# 6 Beschikbaar materiaal

- Lesmateriaal van de betrokken partims
- Leho cursussite
- Versioning: github

# 7 Evaluatie

## 7.1 Puntenverdeling

- productevaluatie: elk deel van de opdracht wordt gequoteerd (zie beschrijving van de opdracht onder punt 2). Het ontbreken of het onvoldoende uitwerken van een belangrijk onderdeel kan reeds een onvoldoende voor het project inhouden.
- peer-assessment: het resultaat van het groepswerk wordt vertaald naar een individueel resultaat volgens de individuele inbreng en medewerking.

## 7.2 Hoe gebeurt de evaluatie?

- Elk deel van de opdracht (analyse, project management en uiteindelijk eindproduct) wordt geëvalueerd volgens specifieke criteria. Tijdens de gemeenschappelijke feedbackmomenten en na tussentijdse deadlines wordt door de begeleiders telkens feedback verzorgd waarvan implementatie noodzakelijk is. De quotering voor het projectmanagementdossier wordt opgenomen als permanente evaluatie bij het partim 'Projectmanagement'.

De evaluaties resulteren in een groepscijfer voor het product.

- Op basis van de deelname aan de verplichte coachingmomenten, een peer-en co-assessment over de competenties, gekoppeld aan een functioneringsgesprek in groep kan het individueel cijfer afwijken van het groepscijfer, zowel in positieve als negatieve zin. Ongewettigde afwezigheid op één of meerdere coachingmomenten kan een uitzetting uit de groep impliceren en een nulquotering. Dezelfde sanctie kan genomen worden indien vanuit de groep voortdurend klachten komen dat één of meerdere studenten niet of onvoldoende meewerken. Uiteraard worden in dit geval alle partijen gehoord vooraleer een beslissing wordt genomen. **Hou er in elk geval rekening mee dat het project niet opnieuw gedaan kan worden in tweede zittijd (zie verder).**

## 7.3 Wat bij niet slagen in eerste zittijd

Gezien het project gecoacht wordt is een herkansing in tweede zitting niet mogelijk. Er worden regelmatige feedbacksessies ingelast waarbij het resultaat tussentijds geëvalueerd wordt. De groep moet uiteraard met deze evaluaties rekening houden zodat de kwaliteit van het eindresultaat in positieve zin evolueert.