

Academie ICT & Creative Technologies

Project Handleiding

Memory

Software ontwikkeling module –periode1



Inhoud

1. Inleiding	3
1.1 Vereiste functionaliteiten Memory-spel	3
1.2 Extra functionaliteiten voor de pro!.....	3
1.3 Groepsindeling	4
1.4 Begeleiding.....	4
1.5 Technieken, tools en talen	4
1.6 Benodigde hardware.....	4
1.7 Benodigde software	4
2. De organisatie	5
2.1 Product owner	5
2.2 Tutor	5
2.3 Afsluiting.....	5
3. Project planning	6
3.1 Voorstel planning	6
4. Op te leveren producten:	8
5. Project beoordelingscriteria	9

1. Inleiding

De module Software Ontwikkeling (Jaar 1, Periode 1) wordt afgesloten met een project. In dit project komen alle competenties geïntegreerd terug die in de colleges aan bod zijn gekomen. Het project toetst de competenties:

- Analyseren
- Adviseren
- Ontwerpen
- Realiseren
- Beheren

op niveau 1 volgens de HBO-I domeinbeschrijving.

In het project toont de student aan over de benodigde kennis, kunde en basisvaardigheden te beschikken die bij deze module behoren.

De student toont dit aan door het ontwikkelen van een software product: het 'memory-spel'. Hierbij gaat het om zo veel mogelijk twee dezelfde kaartjes om te draaien. Voor speluitleg zie:

<http://www.memorykonijn.com/nieuws/spelregels-memory>.

Het project bestaat uit verschillende onderdelen: analyseren en programmeren.

1.1 Vereiste functionaliteiten Memory-spel

In onderstaande lijst staan de eisen aan het memory spel. Dit zijn de

1. Het speelveld bestaat uit 16 memory kaarten, deze zijn als 4 bij 4 kaarten zichtbaar;
2. De achterkant van de kaarten mag zelf gekozen worden.
3. De voorkant van de kaarten mag zelf gekozen worden;
4. Het scherm moet dynamisch (form onload) opgebouwd worden (niet 16 kaarten vooraf op het windows form plaatsen), maar eerst een hoofdmenu o.i.d;
5. Memory moet met 2 spelers gespeeld kunnen worden, beide spelers zijn 'mens' en spelen op hetzelfde bord/scherm (geen meerdere geopende instanties van het memory spel);
6. Bij het starten van het spel ziet de speler alle kaarten 'op de kop' in een 4 bij 4 matrix;
7. De spelers kunnen hun namen ingeven in het scherm;
8. Er moet een knop zijn om het spel te resetten (herstarten).
9. Wanneer het spel is afgelopen, moeten de scores en namen van beide spelers worden opgeslagen en of de speler heeft verloren of gewonnen.
10. Vanuit het hoofdmenu moet er genavigeerd kunnen worden naar een pagina waar de highscores op te zien zijn
11. De huidige status van het spel moet worden vastgelegd in het tekstbestand 'memory.sav' (het spel wordt als het ware bevroren). De volgende aspecten van het spel moeten worden vastgelegd: status van de 16 kaarten, namen van de spelers, scores van de spelers, welke speler aan de beurt is;
12. Terug laden van de opgeslagen bevroren status van het spel. Bij het terug laden van de bevroren status worden de kaarten teruggezet, de namen ingevuld, de scores ingevuld en welke speler aan de beurt was.

1.2 Extra functionaliteiten voor de pro!

Er zijn teams die graag meer willen bouwen dan alleen de vereisten uit de vorige paragraaf. Hieronder

onder staan een aantal extra functionaliteiten, maar ook je product owner kan jullie uitdagen om je Memory Game nóg mooier, beter, aantrekkelijker, sneller en bijzonderder te maken.

- Het toevoegen van sockets voor multiplayer tussen twee computers/laptops.
- Het versleutelen van het memory.sav bestand
- Een grote speelveld
- Muziek & animaties
- Meerdere thema's

1.3 Groepsindeling

Het project wordt uitgevoerd in groepen van 5 studenten. Uitzonderingen op de groepsgrootte in overleg met de projectleiding.

Groepen dienen zich zo snel mogelijk te registreren bij BlackBoard.

1.4 Begeleiding

Alle groepen krijgen een procesbegeleider : de tutor. De groep zoekt op de eerste dag al contact met de betreffende tutor. Tutor en groep spreken elkaar minstens 1 keer per week.

1.5 Technieken, tools en talen

- Scrum/Agile voor softwareontwikkeling
- Versiebeheer met Git/GitHub
- Programmeertaal is C# binnen de ontwikkelomgeving Visual Studio van Microsoft;
- Project is een WPF (*Windows Presentation Foundation*) applicatie zonder koppeling met een database.

1.6 Benodigde hardware

Per student : laptop

1.7 Benodigde software

- Visual Studio (>=2018, zo mogelijk Community edition)
- Trello voor het maken van een scrum bord.
- GIT voor versiebeheer op de software

2. De organisatie

Het project kent naast ca. 160 studenten ook nog tutoren (voornamelijk docenten), experts en studentassistenten.

Projectcoördinatoren: Martin Molema

Product owner(s): Zie BlackBoard Memory project

Project experts:

- Programmeren: Melinda de Roo , Jos Foppele
- Scrum: Harro Medema, Bas van Hensbergen
- Analyse : Harro Medema

Tutor(en): Toegewezen via BlackBoard aan teams.

2.1 Product owner

De product owner fungeert als opdrachtgever voor één of meerdere teams. Na toewijzing van een Product Owner aan een team, neemt het team initiatief om contact te leggen met de Product Owner.

De Product Owner kan binnen de gestelde kaders (zie 1.1) zelf onderdelen aanpassen of uitbreiden om zo aan te sluiten bij de skills van de teamleden. Hij/zij heeft daarmee een belangrijke rol zodat de studenten tot leren kunnen komen en voldoende uitgedaagd worden.

De Product Owner geeft na oplevering van het eindproduct een beoordeling over het eindproduct die meetelt in het eindcijfer van de individuele studenten.

2.2 Tutor

De tutor begeleidt de studenten met het proces van samenwerken en de combinatie 'leren en presteren'. Studenten kunnen moeilijkheden ervaren in het samenwerken binnen het team om uiteenlopende redenen. Indien de tutor de groep niet kan helpen/bijsturen, wordt de module coördinator opgeschakeld.

De tutor geeft na afloop van de module een beoordeling over de teamsamenwerking die meetelt in het eindcijfer van de individuele studenten.

2.3 Afsluiting

De studenten verzamelen zich aan het einde van de module in de kuil om hun product aan elkaar, aan tutoren en Product Owners te tonen.

3. Project planning

In onderstaande diagram staat globaal de planning (schooljaar 2019-2020):

	Datum	2-9	9-9	16-9	23-9	30-9	7-10	14-10	21-10	28-10	4-11
	Week	36	37	38	39	40	41	42	43	44	45
	Periode	P1.1	P1.2	P1.3	P1.4	P1.5	P1.6	P1.7	V	P1.8	T1
Vak		1	2	3	4	5	6	7	8	9	10
Aftrap		Ma 8:30-10:00 t/m woe									
Samenstellen groepen											
Lego Scrum workshop			4								
Requirement Analyse & Use Cases		2	2	2	2	2	2	2		2	
Inleiding programmeren		2	2	2	2	2	2	2		2	
Logica		2	2	2	2	2	2	2		2	
Atelier		17	16	18	18	16	18	18		14	
PPO		2		2	2	2	2	2		2	
Project: programmeren						x	x	x		x	
Project: Pitch + Assessment											Woe
Project: Gezellige afsluiting + demo's											Di (Kuil)
Deadlines											
- Inleveren Blackboard											4-11-2019 23:59
Herkansingen											Vrijdag 9-13u

Voor het schooljaar 2019/2020 geldt:

- 5 november 2019 feestelijke afsluiting en demonstratie
- 6 november 2019 assessments

Het project start met een centrale aftrap in de eerste week op maandag. Daarna worden in de lessen Professioneel Handelen de groepen samengesteld. In periode 1 zijn er niet alleen lessen, maar ook ondersteunende workshops zoals over Scrum en dergelijke.

De studenten werken in de eerste 4 weken aan het ophalen van wensen en eisen, maken kennis met hun tutor en product owner en maken diverse ontwerpen.

Vanaf week 5 gaan de studenten daadwerkelijk coderen in Visual Studio. Het is verstandig om eerder al te beginnen met het opzetten van het project (GIT e.d.), Scrumboard (bijv. Trello) en Visual Studio. Uiteraard kan ook begonnen worden met het uitwerken van bepaalde algoritmes.

De beoordeling vindt plaats aan het einde en bestaat uit 3 onderdelen:

- Verzorgen van een pitch
- Assessment van de code en documentatie
- Demonstratie

3.1 Voorstel planning

Hoewel teams vrij zijn in het opzetten van een eigen planning, is het raadzaam om tijdig te beginnen. Onderstaande planning kan een handvat zijn om de eigen planning mee op te zetten.

- Week 1
 - Groep samenstellen
 - Kennismaking opdrachtgever + tutor
 - Lezen & begrijpen projectbeschrijving
 - Samenwerkingsovereenkomst
- Week 2
 - Requirements verzamelen (interview opdrachtgever)
 - Start ontwerp maken
- Week 3

- Werken aan het ontwerp
 - Afstemmen opdrachtgever
 - Opzetten projectomgeving (GIT, Visual Studio etc.)
- Week 4
 - Start programmeren
 - Ontwerpen afronden
- Week 5
 - Programmeren
- Week 6
 - Programmeren
- Week 7
 - Programmeren
 - Starten documentatie
- Week 8
 - Programmeren
 - Testen
- Week 9
 - Programmeren
 - Oplevering Blackboard
- Week 10
 - Feestelijke afsluiting
 - Assessments
 - Pitches

4. Op te leveren producten:

De volgende zaken moeten in een portfolio worden opgeleverd om het project succesvol af te ronden:

1. De beschrijving van de behoefte van de opdrachtgever.
2. Functioneel ontwerp:
 - use cases
 - user stories
 - mockups/wireframes met verantwoording.
3. Requirementsanalyse en MOSCOW lijst; welke non-functionals zijn belangrijk;
4. Acceptatie criteria: wanneer is het product goed?
5. Bewijs van Scrum sprint backlog en van ieder prototypen aan het eind van de sprint.
6. Testrapportage: monkey test, systeemtest
7. Bewijs voor samenwerking met GIT
8. Software product:
 - De memory game als werkende Windows Applicatie;
 - broncode software: voorzien van commentaar volgens document parser tool van Visual Studio als html pagina's in C#.

Bovenstaande zaken moeten opgeleverd worden in Blackboard **vóór 4 november 2019, 23:59**.

Het doel van het opleveren van de broncode is het kunnen bekijken van het eindresultaat. Daarbij kijken we hoe je als team de code geschreven hebt als input voor het assessment. Tijdens het assessment laat je op **je eigen laptop** een werkende applicatie zien én de broncode.

Bovenstaande bestanden combineer je in één zip-file en lever je in via BlackBoard.

5. Project beoordelingscriteria

De toetsing van het project is opgenomen in de module handleiding. Let op dat ieder teamlid op dezelfde manier beoordeeld wordt! Een taakverdeling binnen het team moet niet leiden tot scheefgroei waarbij bijvoorbeeld één of meerdere teamleden geen bijdrage hebben geleverd aan de broncode van het Memory programma.

Tijdens het assessment ligt de nadruk op het beoordelen van de broncode. Daarbij wordt aan ieder teamlid gevraagd om een stuk code toe te lichten. Indien een teamlid niet meegewerkt heeft aan de broncode of onvoldoende toelichting kan geven, dan vindt een herkansing plaats voor dit teamlid.

De beoordeling van de documenten wordt na oplevering in Blackboard gedaan.