



Datum: _____ Student(en): _____ Beoordelaar: _____

Dealbreakers: <i>if (A && B && C && D) { "cijfer bepalen" } else { "cijfer = 1.0" }</i>					Ja	Nee			
A	is het ontwerp gemaakt en af ?				cijfer bepalen volgens weging hieronder	1,0			
B	is het zelf gemaakt? (<i>authenticiteit vaststellen</i>)					1,0			
C	is het geprogrammeerd en af? (<i>code moet bewijzen dat model werkt</i>)					1,0			
D	is presentatie gemaakt?					1,0			
[80%] Ontwerp					slecht	matig	(ruim) vold	(zeer) goed	totaal
Diagrammen (<i>voldoende aantal, gedetailleerd uitgewerkt</i>)					0	1	3	4	
Architectuur (<i>gescheiden lagen, MVC</i>)					0	1	2	3	
Keuze datastructuur (<i>Gelinkte lijst, logische structuur</i>)					0	1	2	3	
Overerving / Polymorfisme toegepast					0	2	4	6	
rv: 11 max 16									
[10%] Implementatie					slecht	matig	(ruim) vold	(zeer) goed	totaal
Functionaliteit (<i>voor RV of G: uitbreidingen gerealiseerd</i>)					0	1	1,5	2	
rv: 1,5 max 2									
[10%] Vorm (<i>Presentatie</i>)					slecht	matig	(ruim) vold	(zeer) goed	totaal
Opbouw verhaal (<i>samenhangend, helder, logische structuur</i>)					0	1	1,5	2	
rv: 1,5 max 2									
					rv: 14 max 20		/ 2 =		
Opmerkingen									



[80%] Ontwerp	slecht	matig	ruim voldoende	(zeer) goed
Diagrammen	Geen AKD van de uiteindelijke implementatie aanwezig maar enkel DKD.	Wel AKD maar detaillering is te beperkt. Er ontbreken methodes of veel multipliciteiten, relaties etc. Of AKD is RV maar er zijn te weinig Sequence Diagrammen gemaakt of met onvoldoende 'diepte' om het ontwerp te verhelderen.	AKD en SD-en zijn afdoende voor toelichting van ontwerp.	Verschillende versies van AKD qua detaillering (bijvoorbeeld zonder methoden voor structuur en met methoden voor illustreren polymorfisme). SD-en zijn handmatig opgeschoond na genereren zodat ze geen onnodige informatie bevatten.
Architectuur	Geen opdeling in presentation, proces en modellen.	De verschillende verantwoordelijkheden van de lagen zijn over verschillende klassen verdeeld conform juiste gelaagdheid maar de gelaagdheid is impliciet en niet expliciet toegelicht.	Juiste gelaagdheid gebruikt en toegelicht.	Viewcontroller in presentation laag gebruikt. Of Zinnvolle opsplitsing over meerdere controllers in proceslaag voor verbeteren 'separation of concern'
Keuze data-structuur	Geen duidelijke datastructuur of onnodige dubbele administratie. Geen encapsulation van data in een klasse.	Gekozen datastructuur maakt goede toepassing van polymorfisme lastig.	Gekozen datastructuur is bruikbaar voor de oplossing en maakt polymorfisme mogelijk.	Uitdagingen/uitzonderingen in de casus zijn met de gekozen datastructuur slim opgelost.
Overerving / Polymorfisme	Geen overerving gebruikt. Of geen polymorfisme gebruikt.	Polymorfisme is slechts gedeeltelijk gebruikt voor eenvoudigere problemen zoals <i>grafische representatie</i> . Spel-logica is niet middels polymorfisme geïmplementeerd.	Polymorfisme is toegepast om de spellogica te implementeren maar er zijn voorbeelden waarbij te weinig delegation wordt gebruikt.	Polymorfisme en delegation zijn goed toegepast. Alle beslissingen van de spellogica worden genomen in (polymorfe) methodes van klassen die de informatie bevatten om de keuze te kunnen maken. Andere klassen delegeren het maken van de keuzes naar deze klassen.
rv: 11 max 16				
[10%] Implementatie	slecht	matig	(ruim) vold	(zeer) goed
Functionaliteit	Applicatie werkt niet.	Tijdens de testrun blijkt de implementatie buggy te zijn. Of enkel de basisfunctionaliteit van de Sokoban casus is geïmplementeerd.	Een van de twee uitbreidingen is geïmplementeerd.	Beide uitbreidingen zijn geïmplementeerd.
rv: 1,5 max 2				
[10%] Vorm (Presentatie)	slecht	matig	(ruim) vold	(zeer) goed
Opbouw verhaal	Geen presentatie ingeleverd maar losse ontwerpdiagrammen.	Er ontbreekt één van de verplichte onderdelen of dit onderdeel wordt onvoldoende duidelijk toegelicht.	De vereiste onderdelen zijn aanwezig in de presentatie.	De opbouw en structuur van de presentatie is goed. Daarbij is er extra aandacht geschonken aan het toelichten van minder vanzelfsprekende keuzes en hoe deze de uiteindelijke oplossing beter maken.