

Student 1: – Student 2: – ...

Commentaar:

AlgemeenSamenwerking: ☐ blijven samenwerken ☐ werk wordt gelijk verdeeld☐ beide studenten begrijpen de ingediende oplossingBeheersing werktuigen: ☐ gtest ☐ TinyXML ☐ CMake ☐ CLion ☐ Doxygen ☐ GitHub**Commentaar:**

Tests

<input type="radio"/> Nt aanwezig (= geen tests)	<input type="radio"/> Vrkrd. gebruikt (= manuele test)	<input type="radio"/> Beperkt (=nt. alles getest) - domein tests	<input type="radio"/> Voldoende (= invoer) - invoer tests - meerdere fouten	<input type="radio"/> Goed (= uitvoer) - uitvoer tests - scenario's	<input type="radio"/> Excellent (= testcode goed) - modulair - code duplicatie
---	--	---	--	--	---

Commentaar:

Contracten

<input type="radio"/> Nt aanwezig (= geen assert)	<input type="radio"/> Vrkrd. gebruikt (= geen pre-post) - niet in .h file - private/protected	<input type="radio"/> Beperkt (=minimum) - properlyInit	<input type="radio"/> Voldoende (= alleen pre) - getters	<input type="radio"/> Goed (= some pre-post) - setters - output	<input type="radio"/> Excellent (= soms pre-post) - domeinmodel bewegingen
--	---	---	--	--	---

Commentaar:

Plan

<input type="radio"/> Nt aanwezig (= geen planning ingediend)	<input type="radio"/> Vrkrd. gebruikt (= te laat)	<input type="radio"/> Beperkt (=nt. alles)	<input type="radio"/> Voldoende (= alle verplicht)	<input type="radio"/> Goed (= meer) - nt. verplichte - zoals beloofd	<input type="radio"/> Excellent (= bijna alles) - graf. impressie - graphics engine
---	---	---	---	---	--

Commentaar:

Objectgericht Ontwerp en Programmeervaardigheden

<input type="checkbox"/> Basis ontbreekt (= beoordeling niet mogelijk)	<input type="checkbox"/> Basis - code compileert - 0, 1, infinity principe - efficiënt gebruik van datastructuren - code opgesplitst in overzichtelijke files - gebruik van pointers	<input type="checkbox"/> Encapsulatie - private variabelen + getters en setters - functie/ctor parameters beperkt - afschermen van interne datastructuren
<input type="checkbox"/> Domeinmodel - klassen modelleren het domein - klassenamen = substantieven - Gedeeltelijk aanwezig: 1-op-1 overeenkomst tussen klassen en verantwoordelijkheden	<input type="checkbox"/> ADT - functies = acties dat het object onderneemt - functionenamen = werkwoorden - functionaliteit bij data (cohesion) - correct gebruik inheritance	<input type="checkbox"/> Modulariteit (= separation of concerns) - loose coupling - dependency injection - input/output is duidelijk afgescheiden (strings aangemaakt in aparte output klassen) + ook in de testcode

Commentaar: