|  |
| --- |
| Hogeschool Rotterdam |
| INFANL01-2 |
| Practicum |
|  |
| **Practicumcolleges leerplannen: Kevin van Ingen** |
| **11/6/2013** |

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

**Practicumlessen voor INFANL01-2**

### Spelregels

Tijdens de practica besluit de docent om al dan niet een stuk theorie te herhalen. Als de theorie later in de week wordt aangeboden dan de practicumles dan volgt er een korte vooruitblik. Als student werk je ook buiten de practica aan de opdrachten. Je biedt je practica-werk aan tijdens de les om de opdracht ‘afgevinkt’ te krijgen bij je practicumdocent. Het afvinken is voorwaarde om een tentamencijfer op je cijferlijst te krijgen. **Wacht niet** tot de laatste week om je opgaven te laten nakijken. Je docent kan je dan vragen om elke opdracht iets extra’s te doen als compenatie. Practicumopdrachten zijn verplicht. Oefenopgaven niet.

### Inhoud

[Practicumcollege - Les 1 3](#_Toc372883345)

[Practicumopdracht 1A – Problem breakdown en pseudo van je project 5](#_Toc372883346)

[Practicumopdracht 1B: pseudo 5](#_Toc372883347)

[Werkvorm PS (pseudo) 6](#_Toc372883348)

[Practicumcollege - Les 2 7](#_Toc372883349)

[Practicumopdracht 2 (Use-Case en activity diagrammen) 8](#_Toc372883350)

[Practicumcollege - Les 3 9](#_Toc372883351)

[Opdracht 3 (class diagram) 10](#_Toc372883352)

[Practicumcollege - Les 4 11](#_Toc372883353)

[Opdracht 4 (class diagram) 12](#_Toc372883354)

[Practicumcollege - Les 5 13](#_Toc372883355)

[Opdracht 5 (sequence diagram) 14](#_Toc372883356)

[Practicumcollege - Les 6 15](#_Toc372883357)

[Opdracht 6 (state-chart) 16](#_Toc372883358)

[Practicumcollege - Les 7 17](#_Toc372883359)

[Practicumcollege - Les 8 18](#_Toc372883360)

[Oefenopgaven 19](#_Toc372883361)

[Oefening Discover creditcard 19](#_Toc372883362)

[Oefening Tetris 20](#_Toc372883363)

[Oefening Rivercrossing 20](#_Toc372883364)

[Oefening stappen 20](#_Toc372883365)

[Oefening schatzoeken 21](#_Toc372883366)

[Oefening passantensteiger 22](#_Toc372883367)

# Practicumcollege - Les 1

Voorbereiding voor docenten:

* Theorieboek hoofdstuk 1,2
* Modulewijzer lezen
* <http://www.engin.umich.edu/~cre/probsolv/open/first/first.htm>
* Problem hierarchie - The Expected Value of Hierarchical Problem Solving <http://www.cs.toronto.edu/~fbacchus/publications/BYAAAI92.pdf>

Learning Abstraction Hierarchies for Problem Solving <http://www.isi.edu/info-agents/papers/knoblock90-aaai.pdf>

* <http://en.wikipedia.org/wiki/Pseudocode>
* Short-paper <http://www.cs.cornell.edu/Courses/cs482/2003su/handouts/pseudocode.pdf>
* Slides: <http://www.slideshare.net/kiran_kaur/pseudocode-basics> en/of <http://www.slideshare.net/lotlot/pseudocode-3222225> en/of vanaf slide 15 <http://www.slideshare.net/hermiraguilar/algorithm-and-pseudo-codes>
* PPP op <http://www.coderookie.com/2006/tutorial/the-pseudocode-programming-process/>
* Pseudo in een tool <http://www.coderookie.com/2006/tutorial/the-pseudocode-programming-process/>

Leerdoelen

* De student heeft kennis gemaakt met basisvaardigheden voor problemsolving
* De student kan een ‘problem hiërarchy’ maken
* De student kent het verschil tussen analytisch en creatief problemen oplossen
* De student kent pseudo code en kan dit toepassen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 5  10 | Introductie ana 02 practicum  Eventueel herhaling theorie week 1 | Luisteren, vragen stellen  *idem* | Toelichting geven op de cursus en de bijeenkomst  *idem* | Plenair  *idem* | Modulewijzer  Slides week 1 |
| UITVOERN | 35 | Practicumopdracht 1A | Oplossen | Opdracht geven | Projectteam | Werkvorm |
| TERUGKIJKEN | 10 | Terugblik op geleerde | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair | - |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 20  5  10 | Theorie: pseudo code  Goed/slechte pseudo  Theorie: pseudo code | Luisteren, vragen stellen  Opnoemen  Luisteren, vragen stellen | Instructie geven  Activeren antwoordgeven  Instructie geven | Plenair  Plenair  Plenair | Slides  Slides  Slides |
| UITVOERN | 45 | Practicumopdracht 1B | Oplossen | Opdracht geven | Duo | Werkvorm als opdracht |
| TERUGKIJKEN | 5 | Terugblik op les en geleerde | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair |  |

## Practicumopdracht 1A – Problem breakdown en pseudo van je project

Deze oefening doe je met je projectteam. Je hebt veel materiaal gekregen voor deze onderwijsperiode voor het project. Ga aan de slag met je team om je project op te delen in kleinere onderdelen.

Neem de overslag mini-game en maak een onderverdeling van abstracte concepten naar concreet. Beschrijf ook de controle van doorvoer van kratten.

1. Formuleer in 1 zin waar de mini-game over gaat. Brainstorm nu met je team wat er allemaal onder zou kunnen vallen.
2. Doe dit eerst analytisch. Maak een probleem hiërarchie van de game. Probeer onderdelen uit je modulemateriaal te halen en beschrijf waar ze uit bestaan.
3. Het controleren van overslag met een algoritme is niet makkelijk. Bespreek hoe je programmeermatisch een haven van 800\*150 vakken kan ‘aansturen’.   
   Maak hier een PSD van.
4. Lever de diagrammen in bij je practicumdocent.

**Criteria**

* Opgave resulteert in een problem breakdown (hierarchie)
* Hoeveelheid en diepgang
  + Minimaal 4 niveaus diep
  + Minimaal 50 blokjes
* Volledigheid: volgende onderdelen zijn nu beschreven/benoemt:
  + Input en output mogelijkheden
  + Visuele onderdelen
  + Het menu
  + Twee verplichte mini-games
  + Twee eigen mini-games (ligt nog niet helemaal vast) -> brainstorm
* Hiërarchisch correct (concepten geordend van abstract naar concreet)
* PSD:
  + Correcte diagramtechniek (PSD/Nassi-shneiderman diagram)
  + Logisch en werkend algoritme
  + Voldoende diepgang (overslag, beweging kraan, beweging boot, boot in balans houden, juiste gebruik van iteraties en controles).

## Practicumopdracht 1B: pseudo

Maak de pseudocode van de controle en uitvoer van overslag van kratten van de mini-game overslag.

**Criteria:**

* Goed ingesprongen
* Correct hoofdlettergebruik
* Abstracter dan code
* Correct gebruik van loops en controlestructuren
* Algoritme bevat
  + Meerdere type kratten
  + Balans van de boot
  + Volgorde van laden
  + Overladen op trein
* Pseudo vormt goed commentaar van implementatie-code

## Werkvorm PS (pseudo)

Met **pseudocode** wordt over het algemeen een onechte (niet-formele) [programmeertaal](http://nl.wikipedia.org/wiki/Programmeertaal) bedoeld. 'Pseudo' betekent onecht, 'code' verwijst naar de [broncode](http://nl.wikipedia.org/wiki/Broncode) van een [computerprogramma](http://nl.wikipedia.org/wiki/Computerprogramma). Doordat pseudocode een informeel karakter heeft, is het niet geschikt om [gecompileerd](http://nl.wikipedia.org/wiki/Compilatie_(informatica)) en uitgevoerd te worden. Het is hierdoor geen echte programmeertaal. Pseudocode wordt gebruikt om [algoritmen](http://nl.wikipedia.org/wiki/Algoritme) vast te leggen op een door mensen leesbare manier met behoud van de stappen. Natuurlijke taal is hiervoor niet geschikt, omdat deze [ambigu](http://nl.wikipedia.org/wiki/Ambigu%C3%AFteit) is. Een werkelijke (formele) programmeertaal is vaak te strak gedefinieerd en vereist specifieke [kennis](http://nl.wikipedia.org/wiki/Kennis_(wetenschap)) van die taal om het algoritme snel en voor iedereen begrijpelijk te maken. In pseudocode komen bekende constructies uit bestaande programmeertalen voor (zoals [C](http://nl.wikipedia.org/wiki/C_(programmeertaal)), [Java](http://nl.wikipedia.org/wiki/Java_(programmeertaal)) of [BASIC](http://nl.wikipedia.org/wiki/BASIC)). Deze worden echter veel losser gebruikt en vaak in combinatie met geschreven zinnen (vaak in het Engels). Een voorbeeld van pseudocode voor het [sorteeralgoritme](http://nl.wikipedia.org/wiki/Sorteeralgoritme) [Bubblesort](http://nl.wikipedia.org/wiki/Bubblesort) is:

**GET *price* of *item***

**GET *salesTaxRate***

***salesTax* = *price* of *item* times *salesTaxRate***

***finalPrice* = *price* of *item* plus *salesTax***

**DISPLAY *finalPrice***

**END**

Pseudocode is nergens echt gedefinieerd (zoals dat bij het ontwerpen van een echte programmeertaal wel gebeurt), waardoor de schrijver van de pseudocode een zeer grote, haast dichterlijke vrijheid heeft. Hierdoor kan een hoger abstractieniveau worden gebruikt dan in een programmeertaal, hetgeen de leesbaarheid ten goede komt.

### Opdrachten

1. Maak de pseudo voor de PSD voor de val van een tetris blokje.

### Bronnen

http://en.wikipedia.org/wiki/Pseudo\_code

# Practicumcollege - Les 2

Voorbereiding voor docenten:

* Hoofdstuk 1+2
* Introduction <http://www.omg.org/news/meetings/workshops/MDA-SOA-WS_Manual/00-T4_Matthews.pdf>
* Wiki: <http://en.wikipedia.org/wiki/Use_case>

Leerdoelen

* De student kent de basis van UML
* De student kent use-cases en kan deze toepassen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 20 | Eventueel herhaling theorie week 2 | Luisteren, vragen stellen | Toelichting geven op de cursus en de bijeenkomst | Plenair | Slides week 3 |
| UITVOERN | 60  30 | Practicumopdracht 2  Oefening | Oplossen | Opdracht geven | Projectteam  ? | Werkvorm als opdracht |
| TERUGKIJKEN | 10  20 | Terugblik op les en geleerde  Zelfstandig werken | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair | - |

## Practicumopdracht 2 (Use-Case en activity diagrammen)

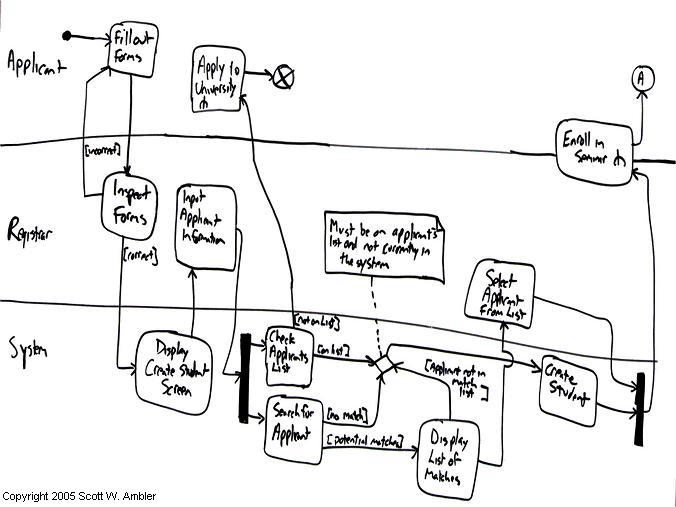
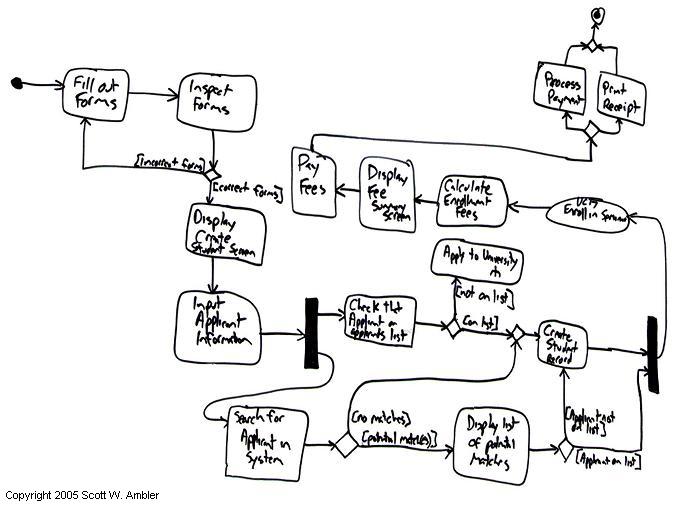
Maak de use-cases die horen bij je spel (maar beperk het tot de overslag mini-game). ‘Lijm’ de use-cases aan elkaar door één activity diagram te maken vanaf het opstarten van het spel tot het einde van de mini-game.

**Criteria**

* Ten aanzien van use-cases:
  + Syntax correct
  + Actor aanwezig
  + Alle use-cases hebben beschrijvingen.
  + Use-case namen bevat werkwoord.
  + Includes gebruikt waar nodig.
  + Alternatieve paden gebruikt waar nodig.
  + Triggers goed aangegeven.
  + Overerving actoren.
  + Feature (‘boundaries’ correct aangegeven.
* Ten aanzien van de activity diagram
  + Bij nesting meerdere diagrammen (anders één diagram)
  + Syntax correct
  + Einde is bereikbaar
  + Actief geschreven activiteiten
  + Beslissingen zijn exclusief.
  + Iteraties zijn eindig
  + Swimlanes gebruikt waar mogelijk (ook AI).
  + Voldoende diepgang (activiteiten zijn niet zinvol verder te splitsen)

### Oefening voor het maken van een activity diagram

1. Maak een activity diagram van een inschrijving op een cursus:
2. Voeg swimlanes toe



# Practicumcollege - Les 3

Voorbereiding voor docenten:

* Hoofdstuk 4 + 5

Leerdoelen

* De student kent class diagrams en kan deze toepassen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 20 | Eventueel herhaling theorie week 3 | Luisteren, vragen stellen | Toelichting geven op de cursus en de bijeenkomst | Plenair | Slides week 5 |
| UITVOERN | 100 | Practicumopgave 3 | Oplossen | Opdracht geven | Duo | Werkvorm als opdracht |
| TERUGKIJKEN | 10  20 | Terugblik op les en geleerde  Zelfstandig werken | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair | - |

## Opdracht 3 (class diagram)

Maak het class diagram wat hoort bij de overslag mini-game. Modelleer alle klassen die niet van Greenfoot zijn (dus die onder de wereld of actor vallen en overige losse classen).

Bij deze opdracht zijn de typen relaties nog van minder belang. De nadruk ligt op correcte classen en attributen.

1. Identificeer alle objecten
2. Identificeer de attributen van de objecten
3. Geef aan welke objecten relaties hebben

Criteria:

* Het class diagram bevat minimaal 6 klassen
* Attributen en methoden zijn beschreven
* Het diagram is gemodelleerd met naleving van de regels uit het boek Learning UML 2.0.
* Het diagram is leesbaar op papierformaat.
* Syntax correct
* Relaties zijn goed weergeven
* Type associaties zijn correct weergeven.
* Gemeenschappelijk gedrag is geabstraheerd.
* Multipliciteit is correct weergegeven.
* Returntypes en argumenten zijn begrijpelijk omschreven.
* Encapsulatie (get/set) zijn gebruikt.
* Modifiers (public, private, etc.) zijn weergegeven voor alle variabelen.
* Class diagrams zijn voorzien van relevante ‘notes’.
* Eventuele afwijkingen van ‘logisch’ design zijn becommentarieerd.

# Practicumcollege - Les 4

Voorbereiding voor docenten:

* Hoofdstuk 4 + 5

Leerdoelen

* De student kent class diagrams en kan deze toepassen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 20 | Eventueel herhaling theorie week 4 | Luisteren, vragen stellen | Toelichting geven op de cursus en de bijeenkomst | Plenair | Slides week 5 |
| UITVOERN | 100 | Practicumopgave 4 | Oplossen | Opdracht geven | Duo | Werkvorm als opdracht |
| TERUGKIJKEN | 10  20 | Terugblik op les en geleerde  Zelfstandig werken | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair | - |

## Opdracht 4 (class diagram)

Maak Het class diagram wat hoort bij de loods mini-game. Modelleer alle klassen die niet van Greenfoot zijn (dus die onder de wereld of actor vallen en overige losse classen).

1. Identificeer alle objecten
2. Identificeer de attributen van de objecten
3. Geef aan welke objecten relaties hebben en van welk type deze zijn
4. Controleer en verbeter ook de relaties van de overslag minigame

Criteria:

* Het class diagram bevat minimaal 6 klassen
* Attributen en methoden zijn beschreven
* Het diagram is gemodelleerd met naleving van de regels uit het boek Learning UML 2.0.
* Het diagram is leesbaar op papierformaat.
* Syntax correct
* Relaties zijn goed weergeven
* Type associaties zijn correct weergeven.
* Gemeenschappelijk gedrag is geabstraheerd.
* Multipliciteit is correct weergegeven.
* Returntypes en argumenten zijn begrijpelijk omschreven.
* Encapsulatie (get/set) zijn gebruikt.
* Modifiers (public, private, etc.) zijn weergegeven voor alle variabelen.
* Class diagrams zijn voorzien van relevante ‘notes’.
* Eventuele afwijkingen van ‘logisch’ design zijn becommentarieerd.

# Practicumcollege - Les 5

Voorbereiding voor docenten:

* Hoofdstuk 7

Leerdoelen

* De student kent sequence diagrams en kan deze toepassen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 20 | Eventueel herhaling theorie week 5 | Luisteren, vragen stellen | Toelichting geven op de cursus en de bijeenkomst | Plenair | Slides week 6 |
| UITVOERN | 100 | Practicumopgave 5 | Oplossen | Opdracht geven | Duo | Werkvorm als opdracht |
| TERUGKIJKEN | 10  20 | Terugblik op les en geleerde  Zelfstandig werken | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair | - |

## Opdracht 5 (sequence diagram)

Maak twee sequence diagrams voor de mini-games overslag en de havenloods van het moeilijkste stuk functionaliteit. Neem de stappen die in één act-methode uitvoer doorlopen worden. Geef ook het bijhorende class diagram (consistent met opgave 3 en 4).

**Belangrijke tip!** Moeilijke stukjes logica kenmerken zich door diepgang van methode calls (elk object roept weer een ander aan), of heel veel methode calls tussen een beperkt aantal objecten. Sequence diagrams maak je als er bij één use-case vier of meer classen nodig zijn en samenwerken of heel veel onderlinge interactie tussen de classen is.

**Criteria:**

* Twee sequence diagrams en twee bijhorende en consistente classe diagrams
* Voor de sequence diagrams
  + Syntax correct
  + Actor aanwezig
  + Consistent met class diagram en use-case (met verwijzing).
  + Call omschreven met mogelijke argumenten.
  + ‘Focus of control’ omschreven.
  + Return-waarde goed omschreven
  + Object instantiaties omschreven.

# Practicumcollege - Les 6

Voorbereiding voor docenten:

* Hoofdstuk 14

Leerdoelen

* De student kent state-charts en kan deze toepassen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 20 | Eventueel herhaling theorie week 6 | Luisteren, vragen stellen | Toelichting geven op de cursus en de bijeenkomst | Plenair | Slides week 7 |
| UITVOERN | 100 | Practicumopgave 6 | Inleveren | Inleveren en aftekenen | Duo | Werkvorm als opdracht |
| TERUGKIJKEN | 10  20 | Terugblik op les en geleerde  Zelfstandig werken | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair | - |

## Opdracht 6 (state-chart)

Maak een state-chart van de twee project mini-games overslag en havenloods. Bij beiden is er sprake van een computergestuurde speler/gedrag. Modelleer het gedrag van deze NPC/AI.

Maak een koppeling van de states in het diagram naar de objecten (of object) uit je classe diagram. Leg via noties de relatie uit.

**Criteria:**

* Twee state-charts
* Syntax correct.
* Start/stop aanwezig.
* Elke staat bereikbaar
* ‘Staat’ omschreven
* Transities goed omschreven.
* Events goed omschreven.
* Relatie naar andere diagrammen verwerkt.
* ‘Activities’ omschreven.
* De koppeling naar het classe diagram is gelegd

**Eventuele bronnen**:

* Voorbeeld persoon: <http://playotherside.com/wp-content/uploads/2011/11/OthersideCharacterStates.png>
* AI in games <http://gram.cs.mcgill.ca/papers/dragert-11-toward.pdf>

# Practicumcollege - Les 7

Voorbereiding voor docenten:

* Herhalen stof

Leerdoelen

* Herhalen alle stof

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fase | Tijd | Inhoud | Activiteiten Cursisten | Activiteiten Docenten | Organisatie | Materiaal |
| VOORUITKIJKEN | 20 | Herhaling theorie | Luisteren, vragen stellen | Toelichting geven op de cursus en de bijeenkomst | Plenair | Slides week 8 |
| UITVOERN | 100 | Inleveren/feedback opdrachten | Inleveren | Inleveren en aftekenen | Duo | Werkvorm als opdracht |
| TERUGKIJKEN | 10  20 | Terugblik op les en geleerde  Zelfstandig werken | Evalueren | Activeren van evaluatie  Optioneel: herhalen van theorie | Plenair | - |

# Practicumcollege - Les 8

Indien de klas in week 2 gestart is, is er geen achtste les. Anders is dit een laatste nakijkmoment.

# Oefenopgaven

Dit zijn enkele oefenopgaven om de *problem solving* vaardigheid te activeren.

## Oefening Discover creditcard

|  |
| --- |
| Alice en Bob moeten met de dreigende overstromingen snel een rubberboot kopen op internet.  Alice wil met haar Discover creditcard betalen, maar kan hem even niet vinden.  Ze weet nog wel een beetje wat voor nummer de kaart had:    Het nummer bestond uit 4 groepjes van vier cijfers  Het eerste groepje was een priemgetal  Het tweede groepje was iets groter dan het eerste en het was een palindroom  Het derde groepje was een kwadraat met maar twee verschillende cijfers  Het vierde groepje was een Fibonacci getal    Samen met Bob kwamen ze er snel achter wat het nummer was en konden ze de boot bestellen! |

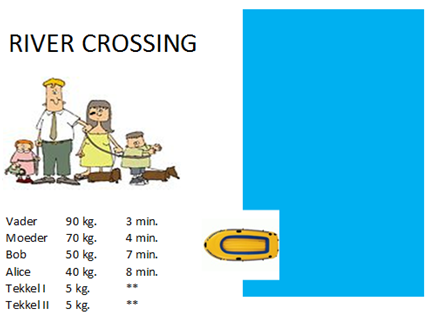
Wat is het nummer van de creditcard?

## Oefening Tetris

Nieuwe spelregel. Regels die vol zijn verdwijnen niet meer. Je verdient punten per ingevuld vakje (1 punt per vakje).

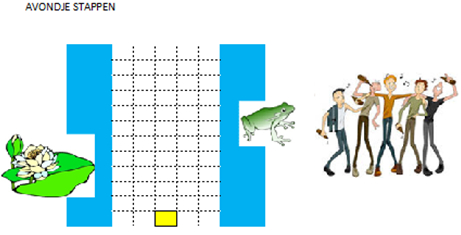
* Wat is de minimale en maximale punten die je nu kunt halen als figuren willekeurig worden aangeboden?
* Wat zijn de minimale en maximale punten als de figuren om-en-om worden aangeboden?

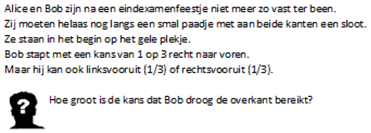
## Oefening Rivercrossing

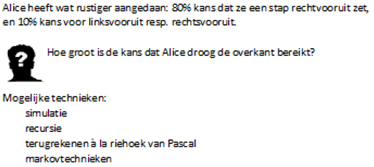




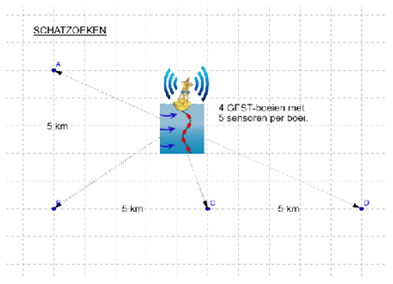
## Oefening stappen

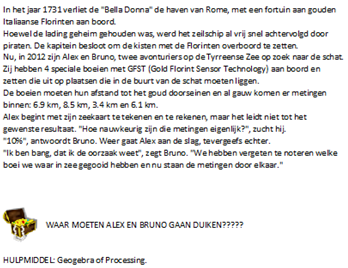




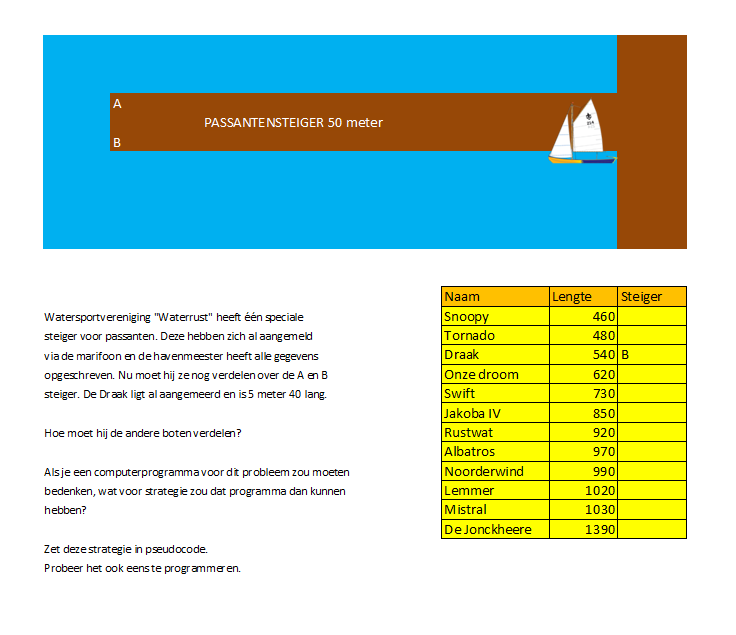


## Oefening schatzoeken





## Oefening passantensteiger



* Maak van deze strategie een PSD

Maak van deze strategie pseudo-code