

Project Java: Richtlijnen

David Blinder

Beerend Ceulemans

Java-Project (zie parallel.vub.ac.be)

Project: een eenvoudige computergame maken.

- Groepjes van 2 of 3 mensen (**niet alleen***)
- Nieuw spelconcept bedenken of variatie op een bestaand spel
- Geen (zuivere) simulatie of kloon van bestaand spel.

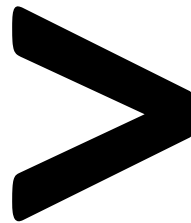
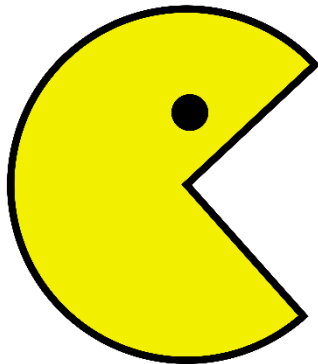
Doel: gebruik maken van klassen als “bouwblokken” voor de spelelementen. Focus op het **programmeren**, niet op de “content”.

- Voorbeelden:
 - Puzzelgames (Bejeweled, Candy Crush, Lasers, enz.)
 - Strategisch (Simcity, turn-based strategy)
 - Arcade games (Petch, Pac-man, enz.)
 - Tower defense, Platformers, Racing. (3D als je een uitdaging wilt)
- Afgeraden:
 - “Klassieke” RPG’s (fetch-quests, grote levels, dialog trees)
 - Uitgebreide leveldesigns (tip: randomized levels)

Je mag bestaande code gebruiken, maar **duidelijk** vermelden

Spelstructuur

Code online hergebruiken is oké (**als je de bronnen vermeld**), maar beperk dit. Doe dit voornamelijk voor dingen die ongerelateerd zijn aan spellogica.



Eenvoudig spel, volledig zelf ontworpen en geschreven

Complex spel, maar bestaat uit een samenraapsel van tientallen online verzamelde codesnippets

“Samenraapsels” neigen eclecticisch en redundant te zijn. Vaak verstaan studenten hun code niet meer goed, waardoor de kwaliteit van de code sterk achteruit gaat.

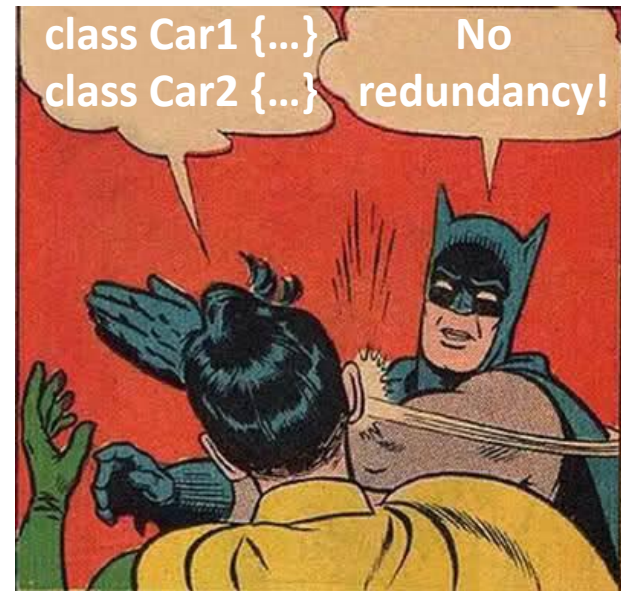
Geen redundantie !

Eén van de belangrijkste metrieken voor codekwaliteit is **geen redundantie!**

Concreet: als je in je code copy-paste gebruikt, doe je bijna altijd iets verkeerd.

Als je code herhaaldelijk moet uitvoeren, maak gebruik van:

- Lussen
- Methodes
- Inheritantie (extends)
- Interfaces (implements)
- ...



Beperk **static** tot een minimum

Beperk het gebruik van het keyword **static** een minimum. In de meeste gevallen is er een betere oplossing beschikbaar.

- Uitzondering: geen probleem bij constanten (**final** keyword)

Soms zal de compiler het gebruik van **static** suggereren. Dit betekent daarom niet dat het een goede designkeuze is.

Kan je geen toegang krijgen tot een (niet-constante) variabele? Mogelijke oplossingen zijn:

- Pas de argumenten aan van de methode waar je de variabele nodig hebt.
- Maak getters voor die parameter
- Verander je klassenstructuur

Minimaliseer de scope van variabelen

Zorg dat variabelen zo lokaal mogelijk gedefinieerd worden.
Bv. als de variabele enkel binnen een methode gebruikt wordt, declareer deze dan binnen de functie zelf.

Declaraties: in volgorde van toenemende scope

- Binnen een lus of geneste klasse
- Methode
- Klasse (private)
- Klasse (public)
- Static (zoveel mogelijk vermijden)

Andere tips

Geen monolieten: reusachtige methodes worden best in kleinere logische eenheden opgebroken

Gebruik constanten en **enums** waar mogelijk. Gebruik logische namen voor je variabelen, bv. **enemy_count** i.p.v. **ec**.

Vermijd gebruik van **instanceof**: zeer zelden de beste oplossing

Becommentarieer je code en maak deze leesbaar (geen 1000 karakters op één lijn, indentatie, enz.)

Kwaliteitsmaat: uitbreidbaarheid. Goede code vergt weinig additioneel werk bij het maken van een extensie (bv. nieuw type vijand/power-up). Slechte code moet volledig herontworpen en herschreven worden.