

Casutoets Object Oriented Programming

Cursus	Object Oriented Programming
Cursuscode	CU75004
Datum	29-01-2020
Begintijd	13:00
Eindtijd	16:00 (recht op 36 minuten extra tijd met faciliteitenpas)
Inleveren	Via inleveringang in Learn

Een andere vriend van je is ook pas aangenomen bij een software-ontwikkelfbedrijf. Dit bedrijf wil zich ook beter in de markt zetten door een game te maken voor kinderen. Echter, niemand in het bedrijf heeft ervaring met het maken van games. Daarom is jouw vriend de opdracht gegeven uit te zoeken hoe game development werkt, een game te bedenken en uit te werken. Dat is gelukt, en je vriend lijkt een game gemaakt te hebben die het bedrijf graag wil verkopen. Er moet alleen nog een scoresysteem worden toegevoegd. Maar daar had je vriend geen rekening mee gehouden. Het toevoegen van de tekst om de score weer te geven is gelukt, maar elke poging om ook maar iets van scorefunctionaliteit aan de game toe te voegen mislukt hopeloos.

Gelukkig heb jij de cursus Object Oriented Programming gevolgd en kan jij wel bedenken wat hier aan de hand is. Je opent de code en inderdaad: vrijwel alles staat in één klasse waarin alles dus afhankelijk is van alles. Na een half uurtje tekenen en nadenken, ben je tot het volgende ontwerp gekomen.



Nu belt je vriend in paniek op: de game moet morgen af zijn. Hij belooft je de helft van zijn salaris als je kan helpen. Kun je het herontwerp implementeren en het scoresysteem gelijk toevoegen?

De game

“Let buzz fly” is een game waarin Buzz Lightyear onderin van links naar rechts beweegt. Dit gebeurt automatisch. Wanneer je op de spatiebaal drukt dan is Buzz in ‘hyperjump’ en kan hij een munt pakken. Als hij in ‘hyperjump’ is dan kan hij niet naar links of rechts bewegen. De munten verschijnen op random posities. Als hij een munt pakt dan levert dit punten op. Als je op het naar-beneden-pijltje drukt gaat Buzz weer uit de ‘hyperjump’ en beweegt hij weer onderaan het scherm van links naar rechts.



Toetsopdracht

Schrijf de gegeven code om volgens het gegeven klassendiagram. Voeg daarbij scorefunctionaliteit toe zoals beschreven in het klassendiagram. De score in de game wordt berekend aan de hand van de volgende regels:

- Het scoren van een muntje levert 3 punten op

LET OP: het kan gebeuren dat je tijdens je implementatie andere keuzes maakt dan het klassendiagram voorschrijft. Dit wordt aangegeven met de drie punten in het klassendiagram. Als het gebeurt dat je implementatie afwijkt van het klassendiagram, leidt dat – mits je uitwerking voldoet aan de beoordelingscriteria – niet tot een slechtere beoordeling.

Zie volgende pagina voor handige tips en de beoordelingscriteria!

Tips:

- Het "npm run watch" commando gaat niet altijd goed om met nieuw toegevoegde klassen. Als je een nieuwe klasse toevoegt, herstart dan de compiler (in de terminal: "ctrl + c" en daarna weer "npm run watch").
- Refactor niet te veel code tegelijk. Controleer na elke refactoring of je game nog doet wat die moet doen.
- Kijk goed naar de verschillen in constructor van de klasse GameItem en zijn subklassen in het klassediagram.

Inleveren

Je levert in Learn een .zip bestand in (geen .rar, .7z, of ander formaat) van maximaal 5mb. Dat .zip bestand bevat uitsluitend de "src/" map met alle code. Alle andere mappen of bestanden lever je niet in.

Beoordelingscriteria

Code die fundamenteel niet compileert, levert geen punten op. Dat wil zeggen, een enkele spelfout of verkeerd haakje wordt door de beoordelaar opgelost, waarna je uitwerking wordt nagekeken. Code waarvoor langer dan een halve minuut moet worden opgelost door de beoordelaar wordt beschouwd als "compileert fundamenteel niet".

Nummer	Leeruitkomst	Aantal punten
1	Je pas op consistente manier indentatie in de code toe	0,2
2	Je voorziet de code volgens een standaardafpraak – AirBnB JavaScript Style Guide – van commentaar	0,4
3	Je geeft consistent betekenisvolle namen aan variabelen, methodes en klassen volgens AirBnB JavaScript Style Guide	0,4
4	Je structureert je code in klassen met attributen en methodes	2
5	Je pas overerving toe om codeduplicatie te voorkomen	2
6	Je geeft datatypes van attributen en parameters, en returnwaarden van methodes expliciet aan. Datatype "any" is niet toegestaan.	2
7	Je pas encapsulation toe in een klasse zodat de data van die klasse op een goede manier wordt weergegeven voor zijn omgeving	1
8	Je past polymorfisme toe opdat code minimaal wordt herhaald	1
9	Je past DRY-principes toe zodat de complexiteit van de code wordt verdeeld in onderhoudbare onderdelen	1