



Programmeren4 PAINTING

Assessment

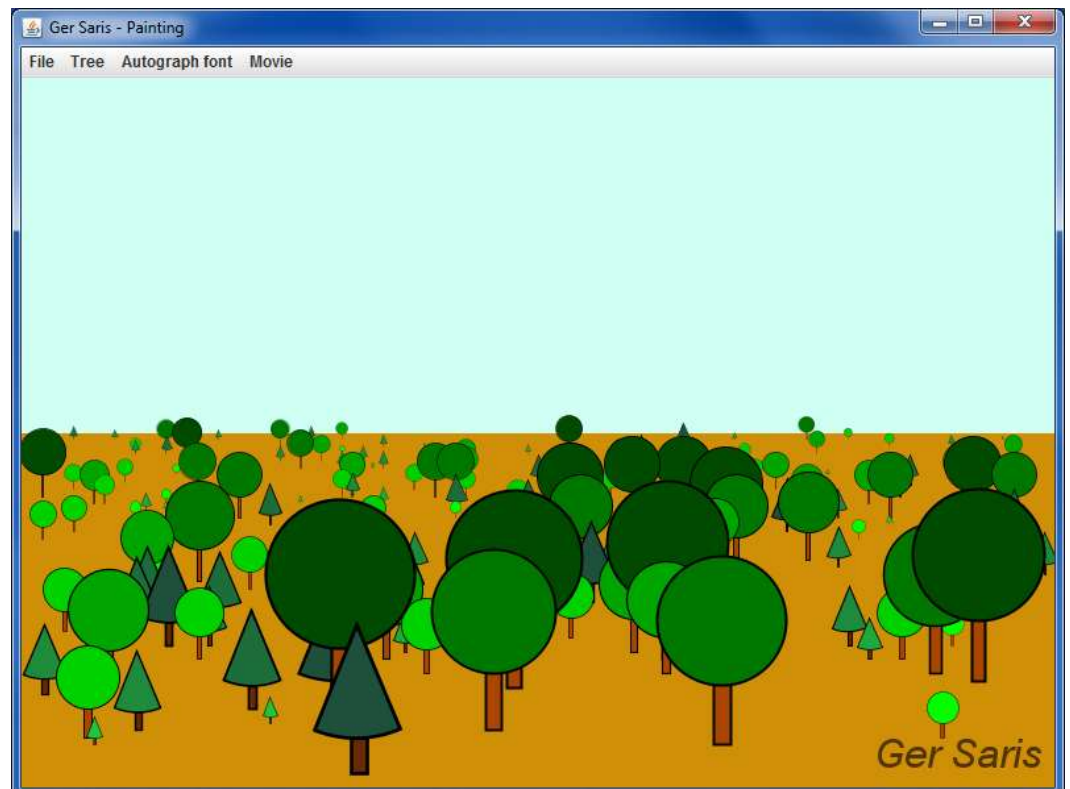
AssessmentRegels

Wij gaan ervan uit dat je alles weet en kunt (gebruiken) wat wij hebben behandeld in het boek, de sheets, de practicum- en de huiswerk-opdrachten van PROG1 tot en met PROG4. Je mag best overleggen met een medestudent, maar je mag geen code delen of voor elkaar schrijven. Als we het vermoeden hebben dat je geen kennis hebt van de (werking van) de code, rapporteren we een fraudemelding bij de examencommissie. Je mag vragen stellen maar het is niet de bedoeling is dat een docent ingaat op jouw uitwerking of meehelpt

*Zo... weer naar uitgekeken zeker 😊 en hier zie je het dan voor je:
De opgave voor het eindassessment. Voor het maken van een uitwerking krijg je een werkweek de tijd en je doet het alleen. De beoordeling vindt plaats door een eindgesprek met een assessor, dus je moet precies weten wat jouw code doet, waarom en hoe je iets snel aan kunt passen. Heel veel succes en plezier!*

Tips

- Verkijk je niet op de complexiteit van deze opdracht. Het lijkt misschien simpel, maar begin op tijd. Je krijgt niet voor niets een volle werkweek.
- Gebruik de meegeleverde assets (plaatjes en geluidjes).
- Op Brightspace staat een link naar een film op MyMedia van de werking van de **BASIS** van de applicatie, die je kunt gebruiken als voorbeeld. Het hoeft niet exact 1-op-1 nagebouwd te worden, gebruik het als voorbeeld.
- Op Brightspace komt een verzameling van Q & A (vragen en antwoorden) te staan over alles waar jullie na het aandachtig bestuderen van dit document nog vragen over hebben. Lees die lijst ook regelmatig door gedurende de looptijd van de opdracht, ook al denk jij geen vragen te hebben, het kan misverstanden voorkomen.



Afbeelding 1: app window

Opdracht**BASIS**

In het kort: Maak de volgende applicatie: Een interactief schilderij met naaldbomen en loofbomen. De gebruiker kan door random bomen toe te voegen het schilderij veranderen. Aan de plaats, kleur en afmeting van de bomen zijn voorwaarden verbonden. De gebruiker kan het schilderij opslaan in een bestand, opgeslagen schilderijen terug inlezen en het schilderij wissen. Het schilderij wordt gesignd met een handtekening in een te kiezen font. Als klap op de vuurpijl is het schilderij in beweging te zetten, waardoor het lijkt alsof je in een trein zit en uit het raam kijkt en het landschap aan je voorbij ziet trekken.

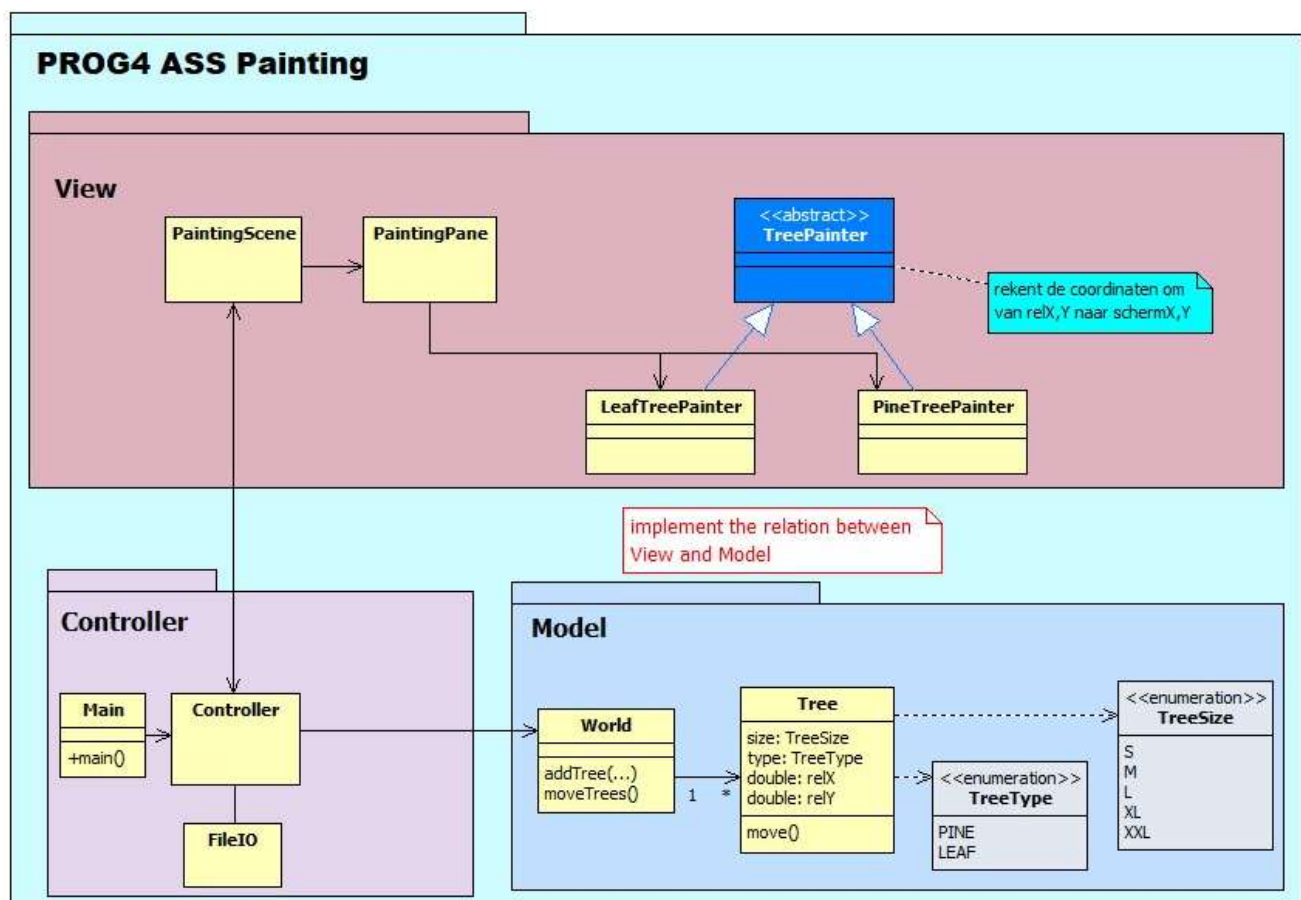
Bekijk de film op Brightspace/MyMedia, want die geeft een helder beeld van de werking van het eindproduct. Je moet dit gaan nabouwen, waarbij je een mate van vrijheid hebt.

Hieronder zie je aan welke eisen jouw werk moet voldoen om een 7 te kunnen halen. Dat heet in dit document verder de **BASIS**.

Voor een hoger cijfer kun je bonusopdrachten maken die de **BASIS** uitbreiden. Voor ieder bonusopdracht zijn de erbij vermelde punten te verdienen.

Klassendiagram

Hieronder zie je een structuur van klassen. (Dit model kun je op Brightspace ook als los plaatje vinden: “[ASS_Painting_KD.jpg](#)”). Het is te gebruiken als leidraad, want dit is een toets over programmeren en we willen jouw modelleervaardigheden niet meten, maar als je liever een eigen ontwerp maakt dan mag dat, zolang je daarbij voldoet aan alle andere eisen.



Afbeelding 2: Klassendiagram

Beschrijving van Eisen**[1] Main**

Maak een class Main met daarin een main(...) method. Instantieer vanuit main(...) een Controller-object

[2] Controller

Deze class regelt de besturing van de applicatie. Wat daar allemaal onder valt moet je zelf invullen. Je kunt dit afleiden uit het klassendiagram, de gevraagde functionaliteit en de technische eisen (zie requirementslijst).

[3] PaintingScene

Maak een class PaintingScene, die overerft van Scene.

Geef het window de title "<jouw naam> - Painting" zoals in het voorbeeld.



Zorg dat het programma afsluit als het window wordt gesloten.

Stel de afmeting van dit window in op: width 800, height 600 (zie [Afbeelding 1](#))

Het window hoeft niet resizable te zijn voor een 7 (daarmee kun je wel een bonuspunt verdienen).

[4] PaintingPane

Dit pane moet je gebruiken om het schilderij in te tekenen.

De bovenste 50% van het pane is de lucht, de onderste 50% is de grond.

Het schilderij wordt gevuld met bomen (Tree) uit de wereld (World) uit het model (zie verderop).

Bomen kunnen alleen maar groeien in de grond (dus niet zweven), dus de stam begint in het onderste helft.

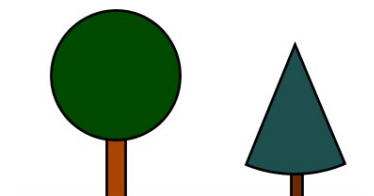
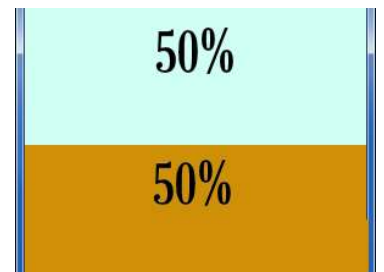
Bomen die dichtbij staan teken je groter, bomen die veraf staan teken je kleiner.

Bomen op de voorgrond staan teken je zichtbaar vòòr de bomen die in de achtergrond staan.

Er zijn twee soorten bomen die je moet kunnen tekenen:

loofboom (LeafTree) en naaldboom (PineTree). Je mag zelf weten hoe je een boom tekent als het maar lijkt op het plaatje hiernaast, dus hoeft niet precies hetzelfde te zijn.

PaintingPane is een weergave van het model World. Bij iedere keer dat paintComponent(...) wordt uitgevoerd wordt de hele wereld opnieuw getekend, de kennis haalt PaintingPane uit het model.

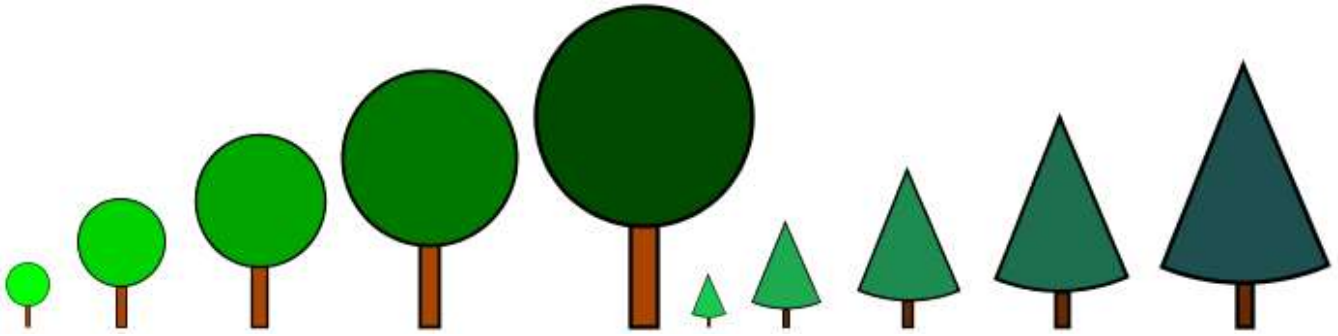
**[5] TreeType**

Is een enum en kan de volgende waarden hebben: LEAF, PINE

[6] TreeSize

Is een enum en kan de volgende waarden hebben: S, M, L, XL, XXL

Bij het tekenen moet je iedere size ten opzichte van elkaar ook in grootte laten verschillen. Om duidelijk zichtbaar onderscheid te kunnen maken moet je aan iedere size ook een unieke kleur geven.



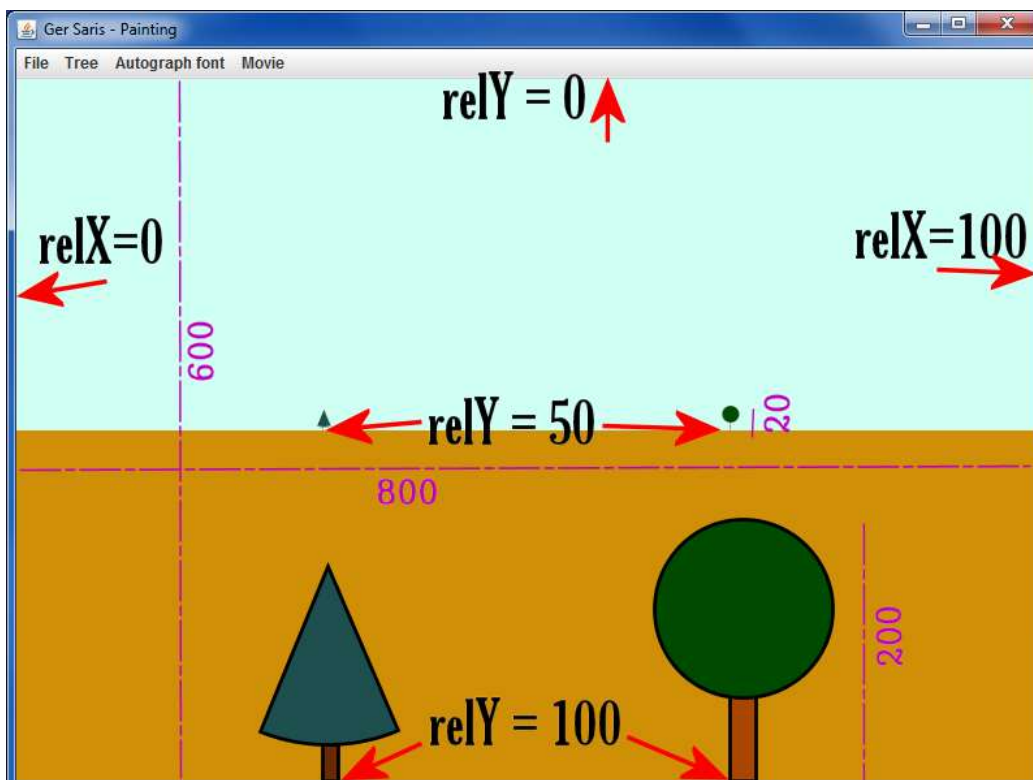
[7] Tree

Dit is een model-klasse en heeft kenmerken: relX, relY, TreeSize, TreeType

Met relX en relY (type double) wordt bedoeld een relatieve positie op het doek die minimaal 0 en maximaal 100 is. Je kunt dit zien als een percentage, waarbij relX de relatieve afstand tot de linkerkant en relY de relatieve afstand tot de bovenkant van het doek aangeeft. De waarde van relY moet minstens 50 zijn.

[8] TreePainter

Is een abstracte class die door de twee classes LeafTreePainter en PineTreePainter concreet wordt ingevuld. In TreePainter worden de relX en relY omgezet naar concrete schermcoördinaten.



```
pine:XXL:30:50
leaf:XXL:70:50
pine:XXL:30:100
leaf:XXL:70:100
```

Je ziet hiernaast twee bomen van afmeting XXL op de onderste lijn, waarbij relY=100, en op de horizon, waarbij relY=50, staan. Het scherm heeft afmeting 800x600 (pix). Die bomen krijgen na omrekening een formaat dat past bij de positie, op de voorgrond 200 (pix) en op de horizon 20 (pix). Bomen van andere TreeSize schalen relatief op dezelfde manier.

Hiernaast zie je twee bomen van size XXL, een op de voorgrond en een op de achtergrond. Ze hebben dezelfde size, maar door de positie worden ze kleiner (achtergrond) of groter (voorgrond) weergegeven. Ze hebben dezelfde kleur, waaraan de size te herkennen is.

[9] LeafTreePainter

Is een overerving van TreePainter en heeft de kennis van hoe een Tree van type LEAF afgebeeld moet worden.

[10] PineTreePainter

Is een overerving van TreePainter en heeft de kennis van hoe een Tree van type PINE afgebeeld moet worden.

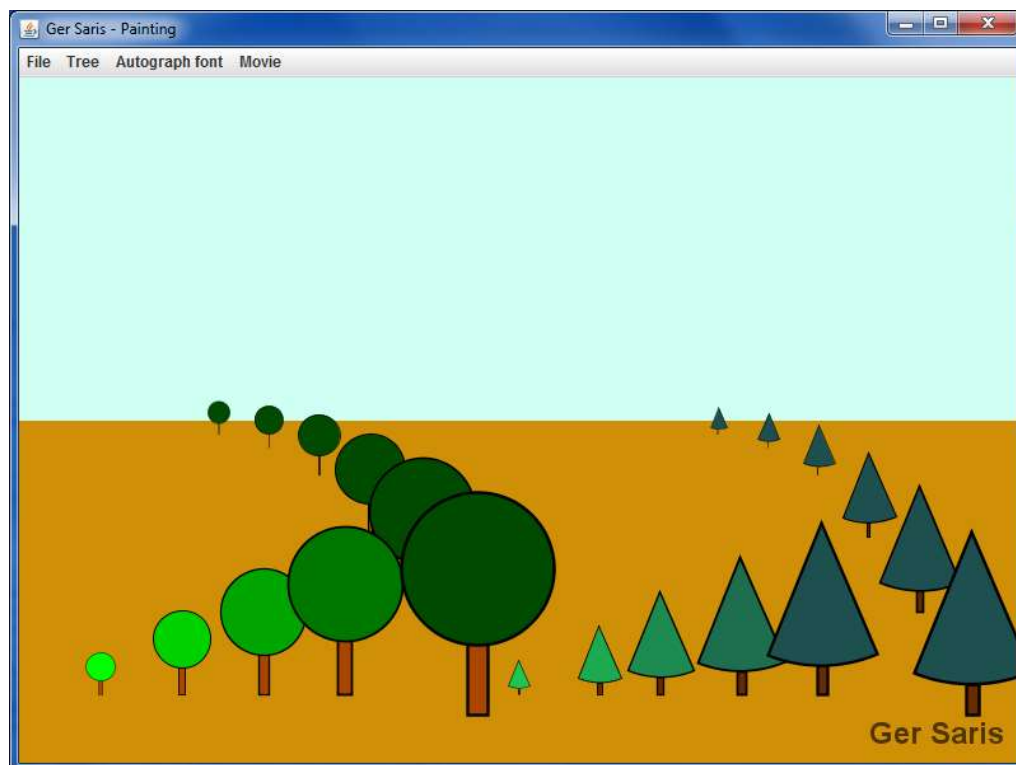
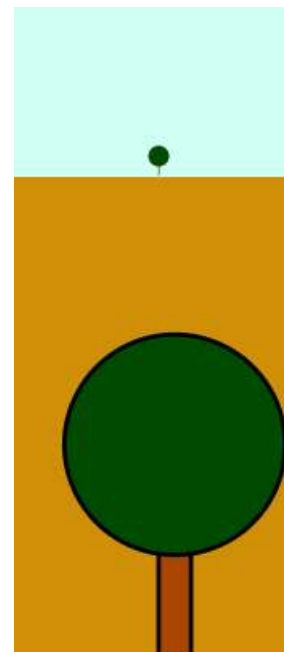
[11] World

Dit is een model-klasse en bevat een verzameling Trees. World heeft een methode om alle bomen een stap te verplaatsen, die nodig is bij de film. Aan de World kunnen bomen worden toegevoegd of vanuit het menu, of alle bomen worden gewist. Als een painting file wordt gelezen wordt de World vervangen of overschreven.

[12] Save & Load Painting

Vanuit het menu moet het mogelijk zijn om een Painting op te slaan als bestand. Gebruik daarvoor een FileChooser waarbij je standaard naar een folder wordt geleid waarin paintings worden opgeslagen. Maak deze folder aan in jouw project en lever die mee in. Plaats de test-painting-bestanden die bij de opdracht als bijlage kwamen in die map. Je mag er van uit gaan dat de bestanden geen fouten bevatten, dus je hoeft geen foutafhandeling te bouwen met meldingen. Het is wel verstandig om bij het constateren van een fout, het inlezen te cancellen.

Hieronder zie je voorbeeldpainting met daarnaast de bijbehorende gegevens in het voorgeschreven file-format.



"Example.painting"

```
leaf:S:8:90
leaf:M:16:90
leaf:L:24:90
leaf:XL:32:90
leaf:XXL:45:93
leaf:XXL:40:78
leaf:XXL:35:67
leaf:XXL:30:58
leaf:XXL:25:54
leaf:XXL:20:52
pine:S:50:90
pine:M:58:90
pine:L:64:90
pine:XL:72:90
pine:XXL:80:90
pine:XXL:95:93
pine:XXL:90:78
pine:XXL:85:67
pine:XXL:80:58
pine:XXL:75:54
pine:XXL:70:52
```


[13] Painting file-format

Op iedere regel informatie van 1 tree, waarbij de waarden worden gescheiden door een dubbelepunt.

<type>:<size>:<relX>:<relY>

Merk op dat de relX en relY hier als gehele getallen worden opgeslagen, dat mag, maar je mag er ook doubles van maken. De bijgeleverde voorbeeldfiles moeten in te lezen zijn.

[14] Menu

File

load painting: de gebruiker kan een painting bestand kiezen en loaden

save painting as: de gebruiker kan de huidige painting opslaan in bestand

exit: de applicatie sluit



Tree

add LEAF Tree: er wordt een LeafTree toegevoegd aan de painting, random size, random position

add PINE Tree: er wordt een PineTree toegevoegd aan de painting, random size, random position

add 100 Trees: er worden 100 Trees toegevoegd aan de painting, random type, random size, random position

clear all Trees: alle bomen verdwijnen uit de painting

LET OP: waar “painting” staat wordt bedoeld, dat het voor de gebruiker zichtbaar wordt, maar natuurlijk worden de trees door jou aan het model World toegevoegd en daarna weergegeven op het scherm.



Movie

Met de checkbox “play” kan de gebruiker de film aan/uit zetten.



[15] Film

Implementeer een Thread die zorgt dat het schilderij met 24 beelden/sec wordt hertekend. De gebruiker kan dit vanuit het menu aan/uit zetten. Verplaats de bomen een klein beetje naar rechts bij iedere tick. Het lijkt dan alsof je uit het raam van een rijdende trein kijkt. Je kunt dit zien in het voorbeeldfilmpje op Brightspace. Er is wel een belangrijke eis: de bomen op de voorgrond verplaatsen relatief meer dan de bomen op de achtergrond. Het bewegen moet soepel gaan.

eisen voor een 7

In de bijlage “[PROG4_assessment_Painting_Requirements.pdf](#)” zie je de lijst met eisen waar je aan moet voldoen. Het is in feite een opsomming van het boven- en onderstaande verhaal. De assessoren gebruiken deze lijst als checklist om vast te stellen of jouw werkstuk aan alle eisen voldoet.

KnockOut (KO)

In de kolom met kolomkop “KO” staat bij sommige eisen de afkorting “KO”. Dat betekent dat als je aan die eis niet voldoet, dat je dan direct een onvoldoende krijgt en de assessor niet verder gaat met de beoordeling. Aan de andere eisen moet ook worden voldaan om een 7 te behalen, daarbij kunnen er strafpunten worden toegekend voor het niet of niet volledig of niet correct voldoen aan een bepaalde eis. Als je onder de 5,5 zakt door strafpunten, is het ook onvoldoende en gaan we niet verder bepalen hoe laag precies, daar is geen tijd voor, je krijgt dan een NVD (niet voldaan).

OO en MVC

Hieronder staat in de kaders nog een samenvatting van deze begrippen, waar jouw werkstuk aan moet voldoen.

OO

Met OO “Object geOriënteerd”-werken, bedoelen we dat je de basisregels gebruikt over het onderbrengen van code in classes waarbij classes begrijpelijk (understandable), herbruikbaar (reusable), onderhoudbaar (maintainable), uitbreidbaar (extendable) en testbaar (testable) zijn, waarbij je encapsulation toepast en zoveel mogelijk redundantie vermijdt. Het liefst zijn classes klein en hebben een duidelijke en beperkte (eenduidige) verantwoordelijkheid. Je past overerving en/of interfaces toe waar mogelijk en nuttig.

Kader 1: OO toelichting

MVC

Met MVC bedoelen we dat je een applicatiestructuur gebruikt gescheiden in drie lagen : Model, View en Controller, waarbij het Model de toestand van het onderwerp (casus) bevat inclusief business-rules en waarbij de View de verantwoordelijkheid heeft om het model af te beelden en te ontsluiten voor de user/actor. De View mag het Model lezen maar niet wijzigen. Het Model kent **niet** de View (of meerdere Views indien aanwezig). Het Model bevat **geen** userinterface-elementen zoals schermcoördinaten, filenamen of plaatjes en heeft ook **geen** functionaliteit om View-onderdelen te creëren. Het enige JavaFX-gerelateerde dat het Model in zich mag hebben zijn properties.

Kader 2: MVC toelichting

PROG1 t/m 4 best practices

Je past de best-practices toe die je geleerd hebt:

- vermijd magic-numbers maar gebruik constants
- methoden en variabelen heten naar wat ze doen en methoden doen naar hoe ze heten (en niets anders)
- hanteer de ‘standaard java coding style’ bij naamgeving van classes, variabelen en methode (CamelCase)
- gebruik bij voorkeur begrijpelijke enums in plaats van strings of ints (of andere geheimtaal)
- voeg comments toe alleen daar waar nuttig en nodig is
- geef je code een nette leesbare overzichtelijke layout
- schrijf alle javacode in het Engels

Kader 3: PROG1 t/m 4 best-practices

Bonuspunten

Als je een van de onderstaande bonuspunten wilt behalen moet je eerst aan alle eisen voor een 7, de **BASIS**, hebben voldaan daarna mag je zelf kiezen welke opdracht(en) je maakt. Zoals je ziet kun je uit 4 bonuspunten kiezen, je hoeft echter maar 3 bonuspunten te halen voor een 10, mits je ze foutloos maakt en aan de technische voorwaarden blijft voldoen. Het maximumcijfer is een 10.

A) Bonus + 0,5 → Resizable maken

Maak het window (de stage) resizable.

De verhouding tussen lucht en grond blijft gelijk (50%-50% van de beschikbare paneruimte, exclusief menubar). Het window is resizable in verticale en horizontale richting.

Als panehoogte gelijk, panebreedte wijzigt: boomgrootte blijft gelijk, horizontale afstand tussen bomen wijzigt

Als panehoogte wijzigt, panebreedte gelijk: boomgrootte blijft gelijk, verticale afstand tussen bomen wijzigt

Indien bonus B ook gebouwd: De handtekening blijft rechtsonder.

Indien bonus E ook gebouwd: De eventuele blijft ook op dezelfde plaats vliegen bij resizen.

B) Bonus + 0,5 → autograph

Plaats jouw naam rechtsonder in het PaintingPane. Zorg daarbij dat je evenveel afstand houdt tot de rechterkant als tot de onderkant van het pane. Bij het wijzigen van het font door de gebruiker, blijft deze afstand constant. De tekst is daarbij altijd leesbaar en geplaatst voor eventuele bomen.

**Autograph font menu**

Voeg aan de menubar een "Autograph font"-menu toe met minstens 3 items. De bedoeling is dat dit radio-buttons zijn, waarvan je er maar 1 tegelijk ingedrukt kunt hebben. De gekozen fontnaam wordt ook direct gebruikt door het bijbehorende font in de handtekening rechtsonder toe te passen. Je mag zelf ttf fonts in het lijstje plaatsen of gebruik maken van de files die meegeleverd zijn bij de opdracht op brightspace. Zorg dat de fonts in de map Resources in je project komen en dat je ze mee inlevert.

**C) Bonus + 1,0 → veranderend landschap**

Maak een steeds veranderend landschap waarbij minstens de volgende landschappen aanwezig moeten zijn: Woestijn, Bergen, Dorp. Bij ieder landschap verschijnen minstens 2 karakteristieke elementen en de achtergrondkleur verandert naar een herkenbare passende kleur. Elementen kunnen zijn: maar je mag ook zelf iets anders bedenken | Woestijn; cactus, oase | Bergen; bergtop, rotssteen | Dorp; huisje, bankje | De elementen zijn zichtbaar als de painting stil staat afhankelijk van waar de user 'zich bevindt' en als de film gaat lopen komen de elementen langzaam in toenemende mate en vervolgens afnemende mate in beeld en komen achtereenvolgens is een cyclus de verschillende landschappen aan bod. De verschijningsnelheid mag je zelf bepalen. De landschapselementen hoeven niet schaalbaar te zijn (mag wel). Implementeer dit op een OO-manier met een aparte class voor ieder object en maak gebruik van overerving.

D) Bonus + 1,0 → drag & drop trees

Maak mogelijk dat de gebruiker ALLEEN ALS DE FILM STIL STAAT, een boom kan verplaatsen door middel van drag & drop. Alle zichtbare bomen moeten verplaatsbaar zijn. De boom wordt daarbij geschaald, dus hoe verder een boom naar "achteren" hoe kleiner hij weergegeven wordt en andersom, net als in de **BASIS**

E) Bonus + 1,0 → animatie eend

Maak de geanimeerde vliegende eend uit het voorbeeld na, kijk daarvoor in het filmpje.

Toon de geanimeerde vliegende eend op het scherm

het animeren verloopt volgens de eerste 2 'rules of animation'.

<startpositie> = horizontaal: is in het midden van het window, verticaal: in het midden van de lucht

de eend vliegt (fladdert) als de film draait

De user kan met de <toetsen> de eend instructies geven

<toetsen> = '←' pijltje naar links, '→' pijltje naar rechts, 'D' letter, '↓'

pijltje naar beneden

← : de eend beweegt langzaam naar links (en fladdert)

→ : de eend beweegt langzaam naar rechts (en fladdert)

↓ : de eend stopt met bewegen op de plaats (en fladdert)

D : de eend verschijnt op de startpositie of verdwijnt van het scherm
bij verschijnen en verdwijnen is het "quack" geluid te horen



einde