

Programmeren4

TagMan

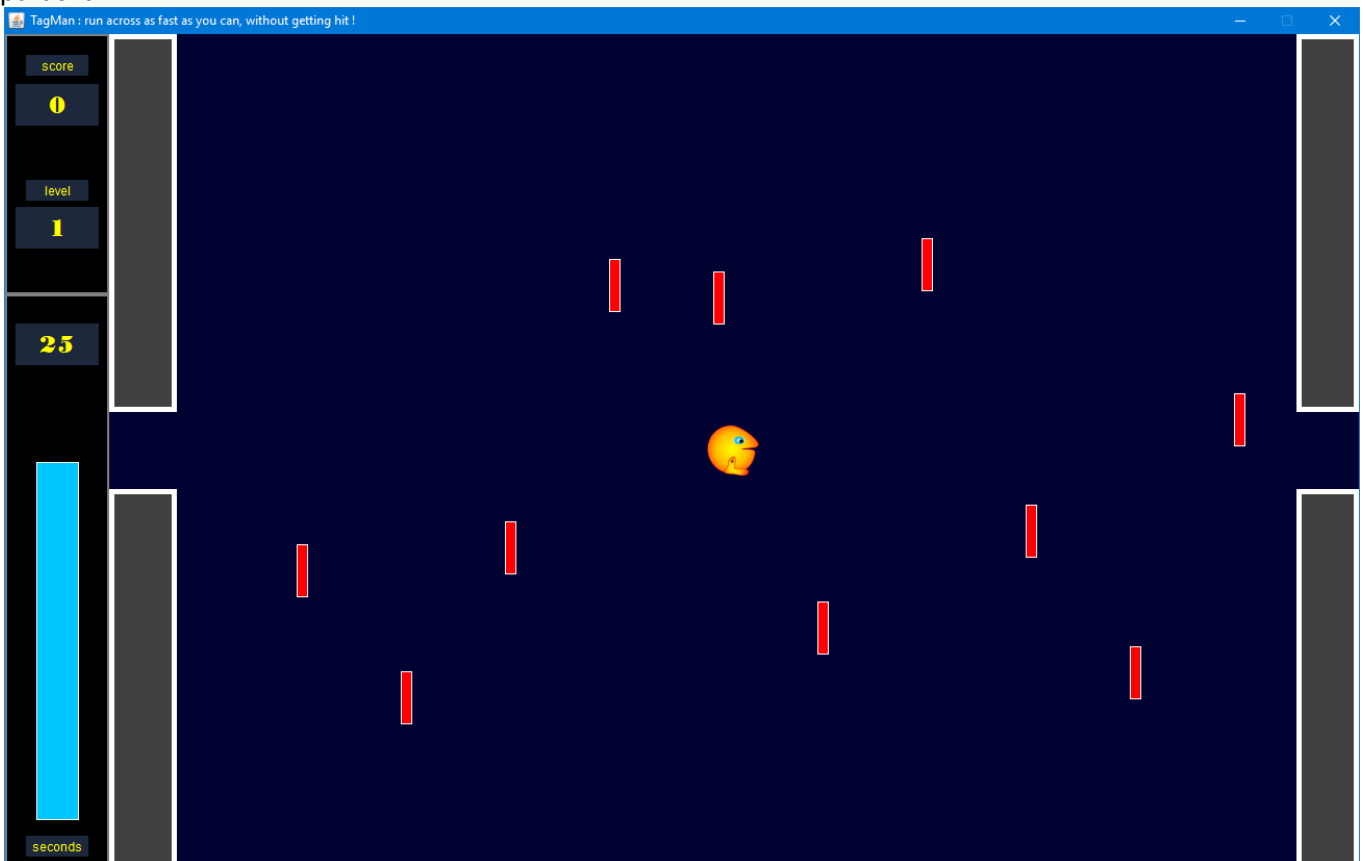
Assessment

AssessmentRegels

Wij gaan ervan uit dat je alles weet en kan (gebruiken) wat wij hebben behandeld in het boek, de sheets, de practicum- en de huiswerk-

opdrachten van PROG1 tot en met PROG4 of VPROG1+VPROG2. Je mag best overleggen met een medestudent, maar weet wel dat jij aan het eind verstand moet hebben van de code die je submit op blackboard. Als een docent het idee heeft dat dit niet het geval is, krijg je een onvoldoende, ook al heb je een fantastisch programma! Tenslotte moet je beseffen dat het niet de bedoeling is dat een docent (een gedeelte van) jouw werk gaat maken. Je kunt dus best algemene vragen stellen, maar geen zaken die rechtstreeks gebonden zijn aan (de uitvoer van) deze opdracht.

Zo... wederom naar uitgekeken natuurlijk en hier ligt het dan voor je: het eindassessment. Voor dit assessment heb je 6 dagen de tijd en maak je alleen. De beoordeling vindt plaats door een eindgesprek met een assessor, dus je moet heel goed weten wat je code doet, waarom en hoe je iets snel aan kan passen. Heel veel succes en plezier!



Tips

- Verkijk je niet op de complexiteit van deze opdracht. Het lijkt heel simpel, maar ... begin op tijd.
- Gebruik de meegeleverde assets (plaatjes), dat scheelt je weer wat tijd.
- Op BB staan filmpjes van de werking van de applicatie, die je kunt gebruiken als voorbeeld.
- Op BB 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.

Opdracht

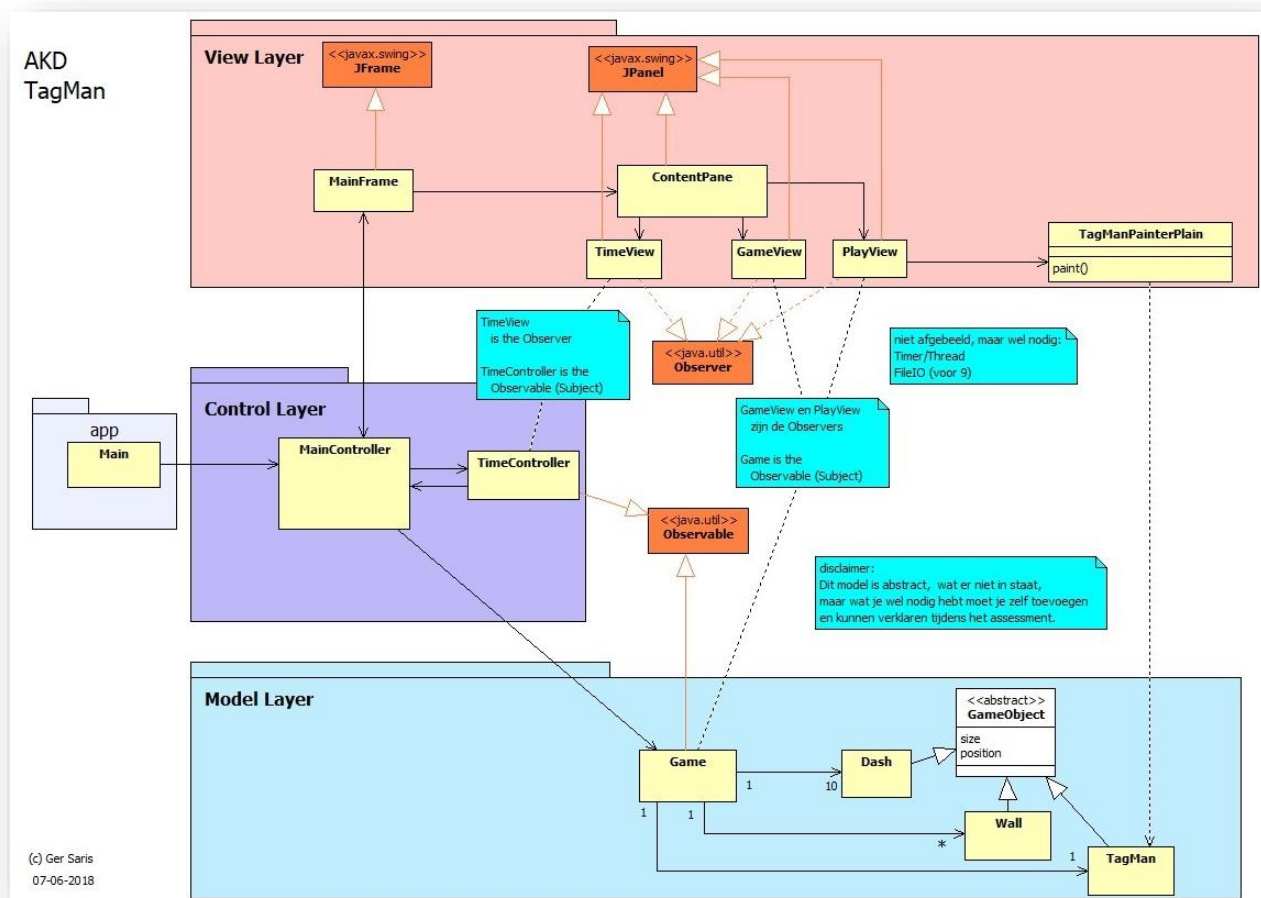
In het kort, maak de volgende applicatie: Een spel waarbij een player zo snel mogelijk van links naar rechts over het scherm moet rennen waarbij vallende staven en muren moeten worden ontweken. De spelscore is het aantal seconden dat je overhoudt per level opgeteld. Als alle levels zijn gespeeld is het spel afgelopen.

Bekijk de films “TagMan_play_for_7_won.gif”, “TagMan_play_for_7_hit.gif”, “TagMan_dont_go.gif”, “TagMan_timer.gif” want die geven 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 maximaal een 7 te kunnen halen. Besef goed dat het niet zo is dat je automatisch een 7 krijgt als je alles hieronder hebt gedaan. Het gaat namelijk om de mate waarin je aan de eisen voldoet. Daarachter staan de eisen waar je aan dient te voldoen om een 8, 9 of 10 te kunnen behalen. Je kunt NIET een tekortkoming uit de eisenlijst voor een 7, compenseren met een feature uit de extra's.

Klassendiagram

Hieronder zie je een structuur van klassen. Je moet dit nabouwen. Disclaimer: dit model is abstract, wat er niet in staat maar wat je wel nodig hebt moet je zelf toevoegen en kunnen verklaren tijdens het assessment. Dit model kun je op blackboard ook als los plaatje vinden, want hieronder is het niet zo goed leesbaar. Als je jouw app in JavaFX bouwt dan moet je de SWING klassen vervangen door JavaFX alternatieven.



EISEN VOOR EEN 7

[1] Main

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

[2] MainController

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] MainFrame

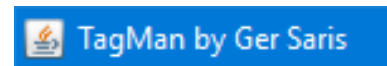
Maak een class MainFrame, die overerft van JFrame.

Geef het window de title "TagMan by <jouw naam>" zoals in het voorbeeld.

Zorg dat het programma afsluit als het window wordt gesloten.

Het window is niet resizable.

In het voorbeeld zijn de schermafmetingen : 1300 width x 800 height. Je mag daarvan afwijken als de essentie maar gehandhaafd blijft.



[4] ContentPane

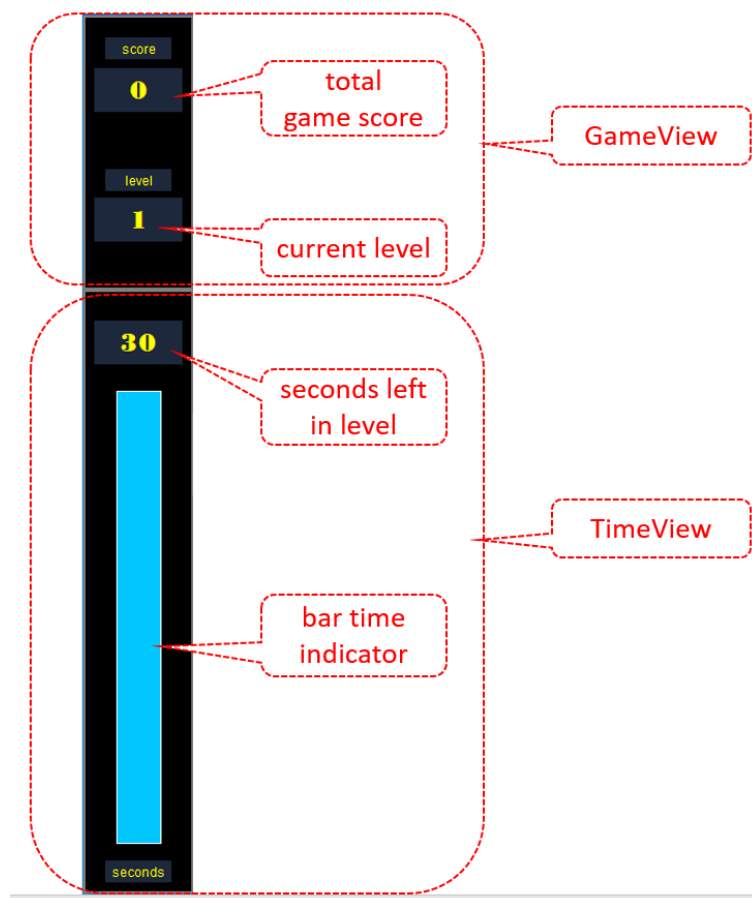
Is een panel dat een als inhoud PlayView, GameView en TimeView krijgt.

[5] PlayView

Dit panel moet je gebruiken om het spel in achtereenvolgende levels weer te geven.



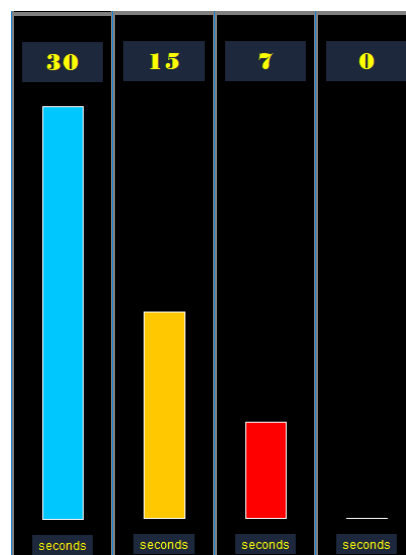
[6] GameView en TimeView



In GameView zijn de “total game score” en het “current level” zichtbaar. Deze twee velden blijven up-to-date gedurende het spelverloop, over meerdere levels.

In TimeView is een “seconds left in level” zichtbaar. De “bar time indicator” geeft visueel weer hoeveel seconden er nog over zijn. De genoemde twee zijn altijd synchroon.

Het totale aantal seconden voor een level is 30. Dan is de bar helemaal gevuld. Iedere seconde neemt de indicator met 1 af. Bij 15 verandert de kleur en bij 7 weer.

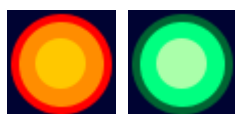


[7] TimeController

Deze klasse bevat de implementatie waarmee je een timer kunt laten aflopen van 30 seconden naar nul. Iedere tick is één seconde. De stand wordt direct zichtbaar in TimeView. Als het spelen van een level start wordt deze timer gestart. Als een level eindigt wordt de timer gestopt. Als de level wisselt wordt de timer gereset.

[8] Player

De klasse TagMan is een model klasse, die de eigenschappen van player bevat. Deze klasse overerft van GameObject. Voor een 7, maak je een niet-animerende player, die je mag tekenen in de vorm van als een cirkel. In de plaatjes hieronder zie je dat de cirkel uit drie kleur banen bestaat. In het plaatje rechts zie je de kleur van de player als hij gefinisht is.



[9] Level spelen

Het programma zet een welkomstboodschap midden op het scherm. Gebruik de key S om de user het spel te laten starten. Gebruik de pijltjestoetsen “up”, “right”, “down” of de toetsen op het numpad 8 (up), 9(right+up), 6(right), 3(right+down), 2(down) om de player te besturen. De player kan het scherm niet verlaten, niet over of door walls lopen. De player kan NIET naar LINKS. De bedoeling is om zo snel mogelijk naar de finish te rennen zonder geraakt te worden door een vallende dash.

Als de player bij de finish aankomt, verandert hij van kleur. Dan toont het programma een boodschap dat de finish is bereikt en hoeveel de level-score is. Met de key L wordt de nieuwe level geladen. Daarna kan de user met de key S het spel starten en herhaalt zich het bovenstaande. Dit gaat door tot alle levels zijn gespeeld. Dat zijn er (voor de 7) twee. Level 1 heeft geen andere muren dan bij de start en finish. Level 2 heeft daarbij nog een muur midden in het scherm. Als je wilt weten hoe dat eruitziet, bekijk dan de filmpjes en bouw dat ongeveer na. Met ongeveer wordt bedoeld dat jouw scherm erop moet lijken maar niet op de millimeter nauwkeurig hetzelfde hoeft te zijn.

Als een speler wordt geraakt door een dash dan is het spel meteen afgelopen. Het programma toont een boodschap “Game Over” met de totaal behaalde score. De key ESC wordt gebruikt om het programma te sluiten.

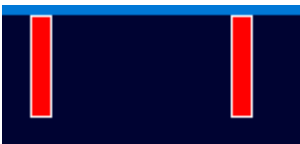
```
tip: boolean java.awt.geom.RectangularShape.intersects(Rectangle2D arg0)
```

De genoemde keys S, L, “pijltjes”, “numpad” mogen uiteraard alleen “in werking” zijn op het bijbehorende logische moment is het spel. Met uitzondering van ESC want die toets is altijd “in werking” en sluit het programma direct, zelfs zonder boodschap.

```
tip: java.awt.event.KeyAdapter
```

```
tip: KeyEvent.VK_ESCAPE
```

[10] Dashes

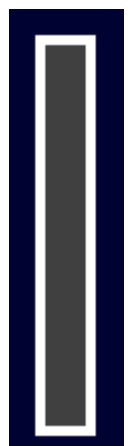


Er zijn tien dashes, die hangen aan het plafond. De dashes komen op willekeurig moment naar beneden, zo worden “losgelaten” maar wel zodanig dat het de player hindert om rechtstreeks naar de finish te lopen. De dashes hebben ieder een willekeurige snelheid maar wel zodanig dat het voor de user zichtbaar is dat ze “vallen”. Hoe je dat oplost in

jouw javacode mag je zelf weten. Ze vallen maar 1 keer per level, als ze eenmaal “gevallen” zijn, blijven ze weg. Ze hebben een vaste afmeting, die mag je zelf bepalen, als het maar lijkt op de afmeting in de filmpjes. De klasse Dash overerft van GameObject.

[11] Wall

De Klasse Wall overerft van GameObject. Iedere Wall heeft daarom zijn eigen size en position. Iedere Wall wordt in PlayView getekend met dezelfde eigenschappen, als een grijze rechthoek met een witte border.



EISEN VOOR EEN 8

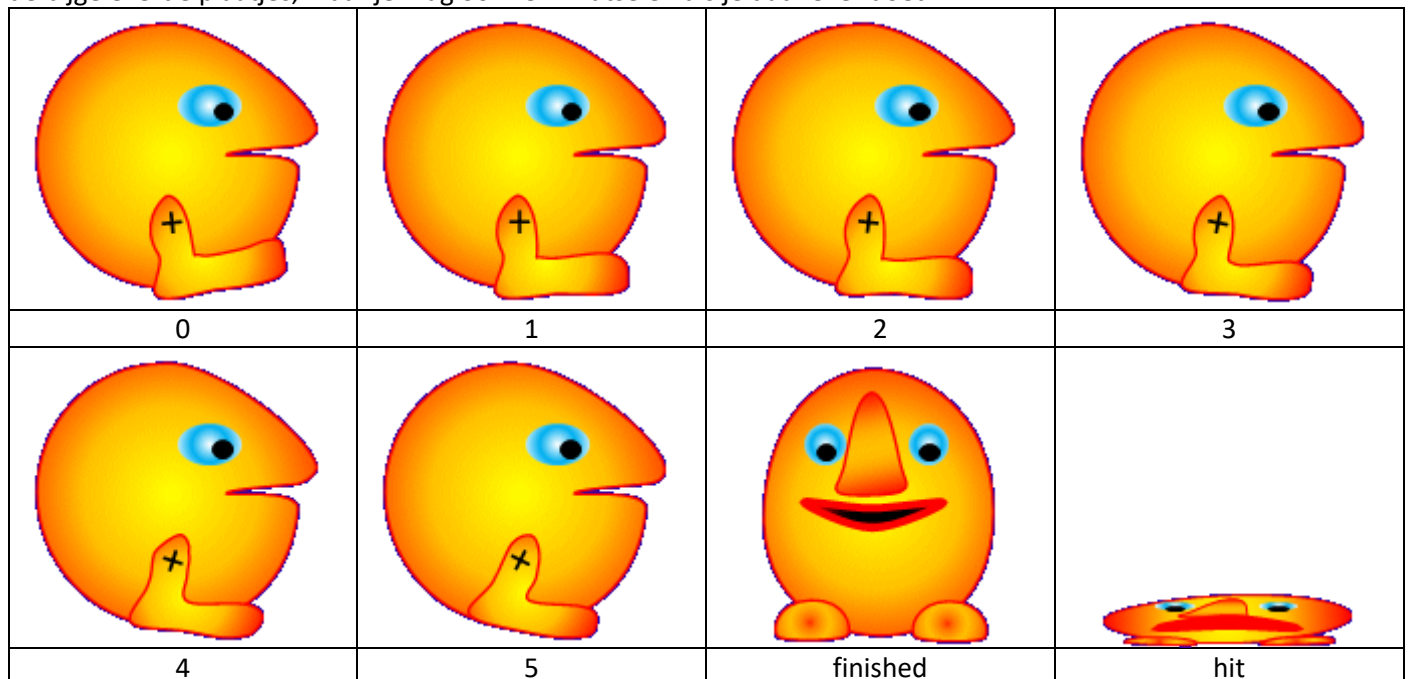
Je hebt aan alle eisen voor een 7 voldaan, anders kun je dit extra punt niet verdienen.
Bekijk de film “[TagMan_play_for_8_won.gif](#)”.

[12] Voeg level 3 en 4 toe

Maak het spel wat uitgebreider door level 3 en 4 toe te voegen. Je mag hiervoor de voorbeelden uit het filmpje nabouwen, maar ook zelf minstens twee muren per level toevoegen. Levels 1 t/m 4 moeten verschillend zijn.

[13] Animatie

Laat de player animeren terwijl hij loopt. Bij iedere stap verandert het uiterlijk een beetje. Je kunt gebruik maken van de bijgeleverde plaatjes, maar je mag ook zelf knutselen als je dat liever doet.



Verander het uiterlijk ook als de player door een dash is geraakt en wanneer hij gefinisht is.

[14] Geluid

Speel een geluid af wanneer: (a) de finish is bereikt en (b) player is hit. Het moeten twee verschillende geluiden zijn en ze mogen niet langer dan 3 sec duren.

[15] ITagManPainter

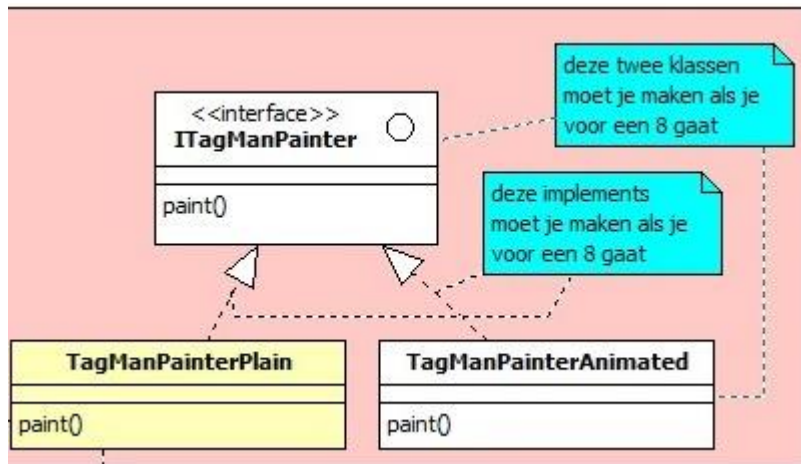
Je moet de code voor het tekenen/weergeven van een player netjes in twee klassen onderbrengen:

TagManPainterPlain en TagManPainterAnimated

De javacode voor de Interface **ITagManPainter** ziet er dan zo uit:

```
public interface ITagManPainter
{
    public void paint(Graphics g, PlayView view, TagMan man);
}
```

Het bijbehorende stukje van het klassendiagram ziet er dan zo uit:



EISEN VOOR EEN 9

[16] Levels in txt files – maken en inladen

Je hebt aan alle eisen voor een 8 voldaan, anders kun je dit extra punt niet verdienen.

Je bedenkt en bouwt een slim formaat om levels in txt bestanden te kunnen opslaan. Daarmee zijn levels buiten de app om te editen met een tekst-editor (bv notepad). De user moet een bestand kunnen kiezen (filechooser). De app moet het txt-bestand kunnen inlezen, een foutmelding geven als corrupte bestand(en) worden aangeboden. In de txt-files staan de indeling, aantal en hoeveelheid van de walls en dashes in het speelveld, en het aantal levels van het spel.

EISEN VOOR EEN 10

[17] Surprise us

Je hebt aan alle eisen voor een 9 voldaan. Daarna mag je helemaal losgaan. Voeg nieuwe <features> toe. Kies maar uit de door ons bedachte opties of bedenk zelf nog iets anders.

<features> = Variatie in dashes

- aantal per level
- bewegingsrichting (van onderen naar boven kan ook)
- aantal keer dat een dash langs komt
- die onderweg van snelheid veranderen

<features> = Scherm dat volledig resizable is terwijl het spel gewoon doorgaat

<features> = Een level-editor waarmee de user zelf levels kan bouwen

<features> = ... (verzin het maar...)

einde