|  |
| --- |
|  |
| Pac-man |
| Fase 3 |
| Gérard lichtert  2019-2020  [Gerard.lichtert@vub.be](mailto:Gerard.lichtert@vub.be)  0557513  Computerwetenschappen |
|  |

|  |
| --- |
|  |
| Wetenschappen & bio-ingenieurwetenschappen |

Inhoudsopgave

[1 Inleiding 3](#_Toc38313871)

[2 Functionele Vereisten 3](#_Toc38313872)

[3 ADT’s 3](#_Toc38313873)

[1. Draw ADT 3](#_Toc38313874)

[2. Grid ADT 4](#_Toc38313875)

[3. Position ADT 5](#_Toc38313876)

[4. Pacman ADT 6](#_Toc38313877)

[5. Coin ADT 10](#_Toc38313878)

[6. Wall ADT 9](#_Toc38313879)

[7. Fruit ADT 12](#_Toc38313880)

[8. level adt 14](#_Toc38313881)

[9. Scoreboard adt 15](#_Toc38313882)

[10. Game adt 16](#_Toc38313883)

[4 Afhankelijkheids Diagram 16](#_Toc38313884)

[5 Planning 17](#_Toc38313885)

[6 log 17](#_Toc38313886)

Inleiding

Dit document bevat de voorstudie van de tweede fase van het spel Pac-Man. We zullen in dit verslag eerste de functionele vereisten bespreken en vervolgens al mijn ADT’s, aanpassingen die ik plan te doen en een korte toelichting van de ADT’s.

Functionele Vereisten

In de eerste fase van het spel is de bedoeling dat het spelwereld weergeven wordt. Het spel moet deels functioneel zijn en de minimale vereisen zijn dat het spel input van de pijlen registreert, en Pac-Man doet bewegen. Munten moeten ook opgegeten worden en er Pac-Man mag niet door muren heen gaan

In de tweede fase van het spel is het de bedoeling dat het spel boven Ade functionele vereisten ook nog een mechanisme krijgt om naar hogere levels te gaan. Bovenop een pacman-adt en munten-adt moet er nu ook een fruit-adt gemaakt worden die het volgende moet kunnen doen: op willekeurige plek op het spelbord een fruit laten verschijnen, het fruit moet verschillende soorten fruit kunnen zijn en zijn respectievelijk 100, 200, 500, 700 en 1000 punten waard. Het fruit-adt mag maar beperkt voorkomen op het spelbord.

Tenslotte moet er ook een scoreboard aangemaakt kunnen worden die het aantal punten bijhoudt, hoogste aantal punten en deze weergeeft op het spelbord.

In de derde fase moeten er ook spookjes komen en ook witte pillen. Ik noem de pillen zelf “powerups”. Pacman moet beiden kunnen eten en de spookjes moeten ook pacman kunnen eten.

ADT’s

1. Draw ADT

**Overzicht**

Het Draw ADT maakt de window aan. Doormidden van de dispatch worden er ook objecten op het scherm getekend en verwijderd.

**Draw ADT**

|  |  |
| --- | --- |
| Draw ADT |  |
| Name: | Signature: |
| Make-draw-adt | ( / 🡪 draw ) |
| Set-key-procedure! | ( any 🡪 / ) |
| Set-game-loop-procedure! | ( any 🡪 / ) |
| Draw-static! | ( static 🡪 / ) |
| Draw-edible! | ( edible 🡪 / ) |
| Dedraw-edible! | ( edible 🡪 / ) |
| Draw-moveable! | ( moveable 🡪 / ) |
| Draw-scoreboard! | ( scoreboard 🡪 / ) |
| Clear-all! | ( / 🡪 / ) |

**Toelichting**

Bij elke draw-static, -edible, -moveable wordt er in de dispatch meegegeven wat voor soort object van de categorie getekend wordt

**Handleiding**

Make-draw-adt maakt het adt aan en maakt een window met de meegegeven parameters.

Set-key-procedure! registreert de invoer van knoppen.

Set-game-loop-procedure! Laat toe dat het spel verderloopt.

Draw-static! Tekent een statisch-adt op het scherm. Mogelijk statische objecten zijn muren en poorten.

Draw-edible! Tekent een eetbaar adt op het scherm. de mogelijke eetbare objecten zijn de powerups, coins en fruit.

Dedraw-edible! Haalt een edible van het scherm af.

Draw-moveable! Tekent een beweegbaar adt op het scherm af. De keuze hieruit is tussen spoken en pacman.

Draw-scoreboard! Tekent het scoreboard doormidden van tiles.

Clear-all! Verwijderd alles van het scherm en zet alle tiles van objecten naar een lege lijst.

1. Grid ADT

**Overzicht**

Het Grid ADT zal posities in een matrix bijhouden, hierdoor kunnen we gemakkelijk elementen zien doen veranderen van positie zonder dat ze buiten het speelgebied gaan.

**Grid ADT**

|  |  |
| --- | --- |
| Grid ADT |  |
| Name: | Signature: |
| Make-grid-adt | ( integer integer 🡪 grid ) |
| peek | (integer integer 🡪 any ) |
| Write! | (integer integer any 🡪 / ) |
| Map-matrix! | (symbol 🡪 / ) |
| Draw-matrix! | (draw-adt 🡪 / ) |
| Count-objects | ( / 🡪 integer ) |

**Toelichting**

Het grid wordt eenmaals gemaakt als een soort map van het speelruimte, in het grid wil ik verschillende speelelementen stoppen doormidden make-rows en make-columns in het level ADT. Hierdoor kunnen we al voorkomen dat het buiten het spel gaat en kunnen we meer elementen in het grid stoppen, verwijderen en aflezen

**Handleiding**

Make-grid-adt verwacht een 2 gehele getallen als dimensiewaardes op een grid of matrix te maken. De matrixelementen zijn standaard allemaal 0. Peek geeft een element in een vector-kotje terug. Vaak zal dit of een 0 zijn of een object-adt. Write! veranderd een vector-kotje destructief. Zowel Peek als Write verwachten gehele getallen en write! verwacht ook nog een extra parameter die gebruikt om dat kotje te set!’en. Map-matrix verwacht een symbool die toepasbaar is op de objecten aanwezig in de matrix en roept deze op bij alle object elementen in de matrix. Draw-matrix! Roept de drawfunctie op bij elk object in de matrix en count-objects telt alle niet 0 elementen in de matrix.

1. Position ADT

**Overzicht**

Het Position ADT maakt een positie object aan

**Position ADT**

|  |  |
| --- | --- |
| Position ADT |  |
| Name: | Signature: |
| Make-position-adt | ( number number 🡪 position ) |
| x | ( / 🡪 number ) |
| Y | ( string 🡪 number ) |
| X! | ( / 🡪 / ) |
| Y! | ( / 🡪 / ) |
| Next-position | ( / 🡪 position ) |
| Compare-position? | ( / 🡪 boolean ) |
|  |  |

**Toelichting**

Position maakt een positie-object waarvan we gemakkelijker x en y waardes van kunnen aflezen alsook wijzigen, posities vergelijken en ook een nieuwe positie maken.

**Handleiding**

Make-position maakt een nieuwe positie met x en y waardes

X en y geven de x en y waardes terug. De x ! en y ! versies veranderen deze waardes destructief. Next-position maakt een nieuwe positie aan maar is NIET destructief. Deze wordt later gebruikt in het level adt om een vergelijking te maken bij het bewegen van pacman

Compare-position? Vergelijkt de positie van het positie-adt met een andere positie die meegegeven moet worden.

1. Static ADT

**Overzicht**

Het static adt maakt statische objecten aan die alleen getekend moeten worden.

**static ADT**

|  |  |
| --- | --- |
| static ADT |  |
| Name: | Signature: |
| Make-static-adt | ( position any 🡪 static ) |
| position | ( / 🡪 position ) |
| Type | ( / 🡪 any ) |
| Draw! | ( static 🡪 / ) |

**Toelichting**

Het is een adt dat niet veel moet kunnen, hoofdzakelijk alleen getekend moet worden de bedoeling is door het maken van een adt en het type mee te geven dat het adt weer wat voor statisch adt het is.

**Handleiding**

Make-static-adt! Verwacht een positie en any als argumenten maar hoofdzakelijk een dit maakt het statische adt aan

Position geeft de position terug

Type geeft het type statische adt weer als symbool.

Draw! Tekent het statisch object op het scherm

1. Edible ADT

**Overzicht**

Het edible adt is heeft weer wat meer functionaliteiten dan het static ADT. Dit adt gebruiken we bij het tekenen van munten, fruit en witte pillen.

**Edible ADT**

|  |  |
| --- | --- |
| Edible ADT |  |
| Name: | Signature: |
| Make-edible-adt | ( position any🡪 edible ) |
| position | ( / 🡪 position ) |
| type | ( / 🡪 any ) |
| Type! | ( any 🡪 / ) |
| On-the-ground? | ( / 🡪 boolean ) |
| score | ( / 🡪 integer ) |
| Set-score-value! | ( integer 🡪 / ) |
| Drop! | ( / 🡪 / ) |
| Eat! | ( / 🡪 / ) |
| Draw! | ( draw 🡪 / ) |
|  |  |

**Toelichting**

Dit adt dient als een bouwsteen voor meerdere eetbare adts

**Handleiding**

Make-edible-adt verwacht een positie en een any argument maar hoofdzakelijk een symbool die aanduid wat voor soort edible het is.

Position geeft de positie terug. Type geeft een variabele terug. Type! Veranderd de variabele type. On-the-ground? Geeft een boolean terug die aanduid of het al niet is opgeraapt. Score geeft de score-waarde van het eetbaar adt. Set-score-value! Wijzigt deze score-waarde. Drop! Zet het eetbaar adt op de grond zodat het weer getekend kan worden. Terwijl eat! Het eetbaar adt weer opeet en het object van het scherm verwijderd.

1. Moveable ADT

**Overzicht**

Het moveable adt maakt een beweegbaar object aan die kan bewegen, van richting veranderen dit adt wordt gebruikt voor het pacman-adt

**Moveable ADT**

|  |  |
| --- | --- |
| Moveable ADT |  |
| Name: | Signature: |
| Make-moveable-adt | ( position any 🡪 moveable ) |
| type | ( / 🡪 any ) |
| Type! | ( any 🡪 / ) |
| Colour | ( / 🡪 any ) |
| Colour! | ( any 🡪 / ) |
| Move! | ( / 🡪 / ) |
| Teleport! | ( / 🡪 / ) |
| score | ( / 🡪 integer) |
| Score! | ( / 🡪 integer ) |
| Position | ( / 🡪 position) |
| direction | ( / 🡪 symbol ) |
| Direction! | ( symbol 🡪 / ) |
| Draw! | ( draw 🡪 / ) |

**Toelichting**

Dit adt heeft veel procedures maar hiervan worden niet alle gebruikt in het pacman adt. Hoewel ze wel gebruikt kunnen worden.

**Handleiding**

Make-moveable-adt maakt een moveable object aan met een positie en het type. Het type kan dan weer teruggegeven worden met type. Type kan weer aangepast worden met type!. De kleur wordt ook bijgehouden in geval het spoken zijn. Pacman houdt daar niks bij. Colour! Dient om de kleur binnen dit adt te weizigen. Dit wordt alleen gebruikt bij het maken van de spook-adts. Move! Beweegt het object stap. Teleport! Teleporteert het object naar de andere kant van het scherm. Dit werkt ook in tegengestelde richting. Score geeft weer de score-waarde terug. Dit kan ook aangepast worden met score! Position geeft de position terug. Direction en direction! Veranderen en geven de richting van het object terug. Tenslotte tekent draw! Het object op het scherm

1. Wall ADT

**Overzicht**

Het wall-adt is een heel eenvoudig adt omdat het gemaakt is van een static-adt.

**wall ADT**

|  |  |
| --- | --- |
| Wall ADT |  |
| Name: | Signature: |
| Make-wall-adt | ( position 🡪 wall ) |

**Toelichting**

Aangezien muren gedetecteerd worden met een simpele (free? Position) procedure in het Level-ADT en daar alleen maar vergeleken word of die positie in het muren grid 0 is of niet moeten we de muren voor de rest alleen maar tekenen. Wat het kan. Alle procedures worden doorverwezen naar de dispatch van het static-adt.

**Handleiding**

Make-wall-adt maakt een muur aan met gegeven positie.

1. Gate ADT

**Overzicht**

Het gate-adt is net zoals het muur adt heel eenvoudig en moet niet meer als getekend worden.

**wall ADT**

|  |  |
| --- | --- |
| Wall ADT |  |
| Name: | Signature: |
| Make-gate-adt | ( position 🡪 gate ) |

**Toelichting**

Er zijn in elk level maar 2 poorten en deze worden net zoals de muren met een simpele (free? Positie) uit de weg gespeeld.

**Handleiding**

Make-wall-adt maakt een poort aan met gegeven positie.

1. Coin ADT

**Overzicht**

Het coin adt verwacht positie en maakt een coin-object aan. Het geeft ook een ‘coin symbool mee wanneer het aangemaakt wordt door midden van een edibles-adt.

**Coin ADT**

|  |  |
| --- | --- |
| Coin ADT |  |
| Name: | Signature: |
| Make-coin-adt | ( position 🡪 coin ) |

**Toelichting**

Het coin-adt is een aanmaak van een edibles-adt met het symbool ‘coin waardoor er een coin getekend kan worden. In dit adt worden alle berichten doorgegeven naar de dispatch van het edibles-adt

**Handleiding**

Make-coin-adt zal een coin aanmaken. Verdere berichten worden opgeroepen met het edibles-adt.

1. Pacman ADT

**Overzicht**

het pacman ADT verwacht een positie om aangemaakt te worden. Het moveable-adt houdt alle gegevens bij.

**Pacman ADT**

|  |  |
| --- | --- |
| Pacman ADT |  |
| Name: | Signature: |
| Make-pacman-adt | ( position 🡪 pacman ) |

**Toelichting**

Dit is het pacman ADT de hoofdpersoon van het spel. Hij wordt bestuurd doormidden van de besturingsknoppen

**Handleiding**

Make-pacman-adt verwacht een positie en maakt een pacman aan doormidden van een moveable-adt. Verder worden alle berichten doorverwezen naar de dispatch van het gemaakte moveable-adt.

1. Fruit ADT

**Overzicht**

Dit ADT maakt een fruit object aan die willekeurig tussen 4 vruchten kiest en ook een score bijhoudt hoeveel elk waard zijn. Het gebruikt een edibles-adt om aangemaakt te worden.

**Fruit ADT**

|  |  |
| --- | --- |
| Fruit ADT |  |
| Name: | Signature: |
| Make-fruit-adt | ( position 🡪 fruit ) |
| Make-a-fruit! | ( integer 🡪 / ) |
| Make-cherry! | ( / 🡪 / ) |
| Make-strawberry! | ( / 🡪 / ) |
| Make-orange! | ( / 🡪 / ) |
| Make-melon | ( / 🡪 / ) |

**Toelichting**

Het fruit moet pas na een tijdje voor een bepaalde tijd op het scherm verschijnen dus wordt het niet direct getekend. In het Level-ADT is er een mechanisme om dit te doen. Het kan opgegeten worden door pacman. Normaal zou er ook verwacht worden dat ook de punten bijgeteld worden maar dit heb ik nog niet kunnen implementeren omdat mijn scoreboard-adt met nummers nog niet volledig op het scherm komt. Het fruit object verschijnt pas na een bepaalde tijd en verdwijnt ook na dezelfde hoeveelheid tijd. Het kan ook opgegeten worden door pacman

**Handleiding**

Make-fruit-adt maakt een fruit aan op positie maar laat die pas na een bepaalde tijd zien.

Make-a-fruit maakt een willekeurige vrucht aan

De make-cherry, make-strawberry zijn de specifieke procedures om specifieke vruchten te maken. Andere berichten worden doorgegeven aan het edibles-adt

1. Ghost ADT

**Overzicht**

Dit adt maakt spoken aan doormidden van een moveable-adt. Het maakt wel verschillende spoken aan.

**Ghost ADT**

|  |  |
| --- | --- |
| Ghost ADT |  |
| Name: | Signature: |
| Make-ghost-adt | ( position 🡪 fruit ) |
| Make-red-ghost! | ( integer 🡪 / ) |
| Make-blue-ghost! | ( / 🡪 / ) |
| Make-orange-ghost! | ( / 🡪 / ) |
| Make-pink-ghost | ( / 🡪 / ) |
| Make-scared-ghost | ( / 🡪 / ) |

**Toelichting**

Het spook adt maakt spoken aan en update ze met de bijhorende situatie (of zij aan het jagen zijn of zij die gejaagd worden)

**Handleiding**

Make-ghost-adt maakt een spook aan. Make-red-ghost, blue, orange en pink maakt hun spook respectievelijk met hun kleur en make-scared-ghost zet ze in een angst staat waardoor ze blauw op het scherm komen. Andere berichten worden doorgegeven aan het edibles-adt

1. Powerup ADT

**Overzicht**

Het powerup-adt is net zoals het coin-adt een aanmaak van een edible-adt en een ‘powerup als type.

**powerup ADT**

|  |  |
| --- | --- |
| Powerup ADT |  |
| Name: | Signature: |
| Make-powerup-adt | ( position 🡪 powerup ) |

**Toelichting**

Alle benodigde procedures bevinden zich in het edibles-adt.

**Handleiding**

Make-powerup-adt zal een powerup aanmaken. Verdere berichten worden opgeroepen met het edibles-adt.

1. level adt

**Overzicht**

Het level adt zal een level aanmaken; maakt alle nodige adts aan en zet ze op het spel. Het dient ook om naar een ander level te gaan.

**level ADT**

|  |  |
| --- | --- |
| level ADT |  |
| Name: | Signature: |
| Make-level-adt | ( integer integer 🡪 Level ) |
| Update! | ( integer 🡪 / ) |
| Draw! | ( draw 🡪 / ) |
| key! | ( symbol 🡪 / or symbol ) |
| Coins-counter | ( / 🡪 integer ) |

**Toelichting**

Dit object maakt een spel-level aan en maakt het spel speelbaar. Het houdt data bij zoals de 2 teleport posities. Het walls-grid, coins-grid het aantal munten dat belangrijk is voor het level-mechanisme. Het pacman-adt en het fruit-adt. Het geeft andere berichten door naar de dispatch van het een scoreboard-adt dat lokaal gemaakt is in het level-adt.

**Handleiding**

Make-level-adt maakt een level object aan met een aantal breedte en hoogte cellen. Update! Update het spel (roept move-pacman en update fruit aan) Draw tekent alle elementen op het scherm. muren pacman coins en fruit. En Key update de key die ingegeven wordt door de speler. Wanneer deze key geet direction is gelijk aan ‘up, ‘down, ‘right of ‘left dan geeft hij de key terug. Coins-ocunter heeft een integer terug die normaal zou moeten gebruikt worden voor het level-mechanisme. Andere berichten worden doorgegeven aan een scoreboard-adt.

1. Scoreboard adt

**Overzicht**

Het scoreboard adt dient om uw huidige score zowel als uw highscore en levens bij te houden.

**Scoreboard ADT**

|  |  |
| --- | --- |
| Scoreboard ADT |  |
| Name: | Signature: |
| Make-scoreboard-adt | ( position symbol 🡪 scoreboard ) |
| score | ( / 🡪 integer ) |
| highscore | ( / 🡪 integer ) |
| lives | ( / 🡪 integer ) |
| Update-score | ( integer 🡪 / ) |
| Add-score! | ( integer 🡪 / ) |
| Update-highscore! | ( integer 🡪 / ) |
| Update-lives! | ( integer 🡪 / ) |
| Reset-score! | ( / 🡪 / ) |
| Draw! | ( scoreboard 🡪 / ) |

**Toelichting**

Score, highscore en levens worden getekend volgens het scoreboard adt in het level-adt. Het scoreboard in het game-adt update steeds de nieuwgemaakte scoreboard in het level-adt. Higschore wordt wel niet dynamisch geupdate zoals levens en score.

**Handleiding**

Make-scoreboard-adt maakt een scoreboard aan die score, highscore en levens bijhoudt. Deze kunnen ook teruggegeven worden doormiddel van score, highscore en lives. Score wordt geupdate met een andere score wanneer een level-adt scoreboard opnieuw aangemaakt wordt. Add-score! Voegt score toe aan de huidige score. Update-highscore! Vergelijkt de huidige score en bind de highscore aan de score op dat moment als het hoger is dan de huidige highscore.

Update-lives! Veranderd het aantal levens. Reset-score! Is om de score terug op 0 te zetten en draw! Tekent het scoreboard op het scherm.

1. Game adt

**Overzicht**

Het game-adt zet het spel in gang wanneer ‘start in de dispatch geroepen wordt. Het maakt een level en teken adt aan zowel als een scoreboard-adt.

**Game ADT**

|  |  |
| --- | --- |
| Game ADT |  |
| Name: | Signature: |
| Make-game-adt | ( / 🡪 game ) |
| start | ( / 🡪 / ) |
|  |  |

**Toelichting**

Dit adt maakt het spel en bevat het levellingsmechanisme. Het houdt ook een scoreboard bij die elke keer wordt meegegeven wanneer de levels veranderen.

**Handleiding**

Make-game-adt verwacht niks en maakt het game-adt.

Start zet het spellus in werking.

Afhankelijkheids Diagram

Position ADT

Static ADT

Edible ADT

moveable ADT

ghost ADT

Grid ADT

Pacman ADT

Powerup ADT

coin ADT

Fruit ADT

Wall ADT

Gate ADT

Scoreboard ADT

Level ADT

Draw ADT

Game ADT

Planning

|  |  |
| --- | --- |
| Week 18 | Aanpassen fase 2 implementaties |
| Week 19 | Aanpassen fase 2 implementaties |
| Week 20 | Aanpassen fase 2 implementaties |
| Week 20 | Powerup-adt gemaakt |
| Week 20 | Ghost-adt gemaakt |
| Week 20 | Gamelogica gemaakt |
| Week 21 | Verslag maken & inleveren |

logboek

|  |  |
| --- | --- |
| Week 18 | Aanpassen fase 2 implementaties |
| Week 19 | Aanpassen fase 2 implementaties |
| Week 20 | Powerup-adt gemaakt |
| Week 20 | Ghost-adt gemaakt & spellogica aangepast |
| Week 20 | Spellogica aangepast |
| Week 20 | Spellogica aangepast |
| Week 21 | Verslag inleveren |