

Programmeerproject 1

Verslag fase 2 “Tower Defence”

Rune Heus 0600234

rune.heus@vub.be

Bachelor Computerwetenschappen 2022-2023

Inhoudsopgave

1. Introductie	2
2. ADTs	2
a) Game ADT	2
b) Environment ADT	2
c) Path ADT	3
d) Tower ADT	4
e) Position ADT	5
f) Monster ADT	6
g) Draw ADT	7
h) Projectile ADT	8
i) Wave ADT	9
j) Player	10
k) Power-up	10
l) Menu	11
3. Afhankelijkheidsdiagram	12
4. Tijdschema	13
5. Instructies van het spel	13
6. Fouten en ontbrekende functionaliteiten	16

1. Introductie

Dit document beschrijft de verschillende abstracte data types en de onderlinge relaties die gecreëerd zijn voor het “Tower Defence” spel van programmeerproject 1 fase 2.

Voor het grafische deel van dit project wordt gebruik gemaakt van een voor-gedefinieerde codebibliotheek van de VUB.

2. ADTs

a) Game ADT

Het Game ADT zal alle functionaliteiten bevatten voor het starten, stoppen en herstarten van het spel. Deze ADT bevat een procedure dat de oproep zal doen voor het inladen van de spelwereld.

Naam	Signatuur	Beschrijving
make-game	(Environment -> Game)	Constructor
clean!	(boolean -> \emptyset)	Zal alle nodige ADT's resetten. De meegegeven boolean geeft mee of het de eerste keer is dat deze gereset wordt of niet.
start!	(\emptyset -> \emptyset)	Zal het begin scherm tonen en wachten tot de gebruiker het spel start.

b) Environment ADT

Het Environment ADT zal voor alle veranderingen in het spel zorgen. Het zal zorgen voor het oproepen van het tekenen van objecten en het verplaatsten ervan. Het ADT zal de informatie bijhouden welke torens en monsters en obstakel er zich bevinden in het spel. Deze ADT kan ook monsters en torens toevoegen en verwijderen.

make-environment	(Teken, Path, Position -> Environment)	Constructor
draw	Draw	Geeft het draw ADT terug.
get-monsters	List	Geeft de lijst terug van alle monsters in het spel.
set-monster!	(list -> \emptyset)	Zal de lijst van monsters veranderen naar de meegegeven lijst.
set-towers!	(list -> \emptyset)	Zal de lijst van torens gelijkzetten aan de meegegeven waarde.
get-path	Path	Geeft het path ADT terug.
add-entity!	(Monster or Tower -> \emptyset)	Zal de meegegeven monster of toren toevoegen aan het spel.
Monsters-process	(\emptyset -> \emptyset)	Een procedure die zorgt voor het process van alles wat een monster moet doen in na 1 frame.

check-tower-areas	(Tower -> \emptyset)	Procedure die in de omgeving van de meegegeven toren zal kijken of er een monster aanwezig is.
on-path?	(Tower -> Boolean)	Zal kijken of de meegeven toren op het pad zou geplaatst worden of niet.
towers-process	(Number -> \emptyset)	Zal het process van de torens starten voor elke frame.
draw-entity!	(Entity -> \emptyset)	Zal via het draw ADT de meegegeven entity teken op het scherm.
find-tower-by-projectile	(Position -> Tower)	Geeft de toren terug van wie het projectiel is.
remove-all-objects!	(\emptyset -> \emptyset)	Zal alle objecten in het spel verwijderen.
monster-random-event	(\emptyset -> \emptyset)	Gaat het random event van een monster uitvoeren als deze er een heeft.
add-obstacle	(Obstacle -> \emptyset)	Voegt een obstacle toe en gaat via het Draw ADT het obstacle tekenen op het scherm.
set-obstacles!	(List -> \emptyset)	Veranderd de obstacles lijst naar de meegegeven lijst.
get-obstacles	List	Geeft de obstacles lijst terug.
reset!	(\emptyset -> \emptyset)	Gaat all objecten in de Environment verwijderen.
obstacle-process	(Number -> \emptyset)	Zal het process van elke obstacle in de lijst uitvoeren.
remove-all-projectiles!	(\emptyset -> \emptyset)	Gaat alle projectielen verwijderen van het scherm en de lijst.
free-position?	(Tower -> Boolean)	Gaat teruggeven of de meegegeven toren geplaatst mag worden of niet.

c) Path ADT

Het Path ADT bevat alle belangrijke posities van het pad, zoals de start en eind positie. Het bevat een lijst die alle draai punten bevat van het pad. Het Path ADT laat ook zien of een monster het einde heeft bereikt.

Naam	Signatuur	Beschrijving
make-path	(Position, Position -> Path)	Constructor
start-position	Position	Geeft de start positie van het pad terug.
end-position	Position	Geeft de eind positie van het pad terug.
path-positions	Lijst	Geeft een lijst terug met alle hoek posities van het pad.
endpoint?	(Monster -> Boolean)	Geeft #t of #f terug of het monster aan het eind van het pad is.

next-path-to-pos	(Position -> Position)	Geeft de volgende path hoek terug vanuit de meegegeven positie.
next-pos-to-far?	(Position, Position -> Boolean)	Geeft #t of #f terug als bij de volgende stap vanuit de eerste meegegeven positie deze verder is dan de 2 ^{de} meegegeven positie.
on-path?	(Object -> Boolean)	Gaat teruggeven of het meegegeven object zich op het pad bevindt.
get-path1	Position	Geeft de positie van path1 terug.
get-path2	Position	Geeft de positie van path2 terug.
random-pos-on-path	Position	Geeft een random positie op het pad terug.

d) Tower ADT

Het Tower ADT bevat alles wat te maken heeft met het aanvallen van monsters. De torens kunnen geplaatst worden in het spel en zullen alle monsters die in de buurt komen aanvallen met projectielen. Een toren kan maar 1 doelwit tegelijk hebben en dit monster zal geselecteerd worden als het in het bereik komt van de toren en deze nog geen doelwit heeft. Elke toren heeft een cooldown dat ervoor zorgt dat deze niet tegen een oneindige snelheid kan schieten.

Naam	Signatuur	Beschrijving
make-tower	(Position-> Toren)	Constructor
get-position	Position	Geeft de positie van de toren terug.
set-cooldown!	(Nummer -> \emptyset)	Zal de cooldown gelijkzetten aan het meegegeven nummer (in seconden).
cooldown	Nummer	Geeft de resterende cooldown terug in seconden.
target	Monster	Geeft het monster terug waar de toren op gefocust is. Als er geen monster gefocust wordt door de toren zal er #f (false) terug gegeven worden.
remove-target!	\emptyset	Zal de target gelijkzetten aan false (#f)
get-projectile	Position	Geeft de positie van een projectiel terug.
entity?	'tower	Geeft 'tower als identificatie.
check-area	(Monster -> \emptyset)	Procedure die zal kijken of het meegegeven monster aanwezig is in de omgeving van een toren.
set-projectile!	(Value -> \emptyset)	Zal het projectiel gelijkzetten aan de meegegeven waarde.
shoot!	(\emptyset -> \emptyset)	Procedure die als hij een target heeft, een projectiel zal afvuren naar dit target.

get-tile	Tile	Geeft de tile terug van de toren.
check-area?	Boolean	Geeft een boolean terug of de toren in zijn omgeving moet checken voor monsters.
get-cost	Number	Geeft de prijs van de toren terug.
get-type	Type	Geeft het type van de toren terug.
get-environment	Environment	Geeft de Environment ADT terug.
remove-all-projectiles	($\emptyset \rightarrow \emptyset$)	Gaat alle projectielen verwijderen van die toren.
remove-projectile-from-list	(Projectile $\rightarrow \emptyset$)	Verwijdert het meegegeven projectiel uit de lijst van projectielen.

e) Position ADT

Het Position ADT is alles wat te maken heeft met het positioneren van objecten in het spel.

Naam	Signatuur	Beschrijving
make-position	(Nummer, Nummer \rightarrow Position)	Constructor
get-x	Nummer	Geeft x coördinaat terug.
get-y	Nummer	Geeft y coördinaat terug.
set-x!	(Nummer $\rightarrow \emptyset$)	Zal de x coördinaat veranderen door het getal.
set-y!	(Nummer $\rightarrow \emptyset$)	Zal de y coördinaat veranderen door het getal.
equal?	(Position \rightarrow boolean)	Predicaat dat zal teruggeven of zijn eigen positie gelijk is aan de meegegeven positie.
change-coordinates!	(Nummer, Nummer $\rightarrow \emptyset$)	Zal de x en y coördinaat veranderen.
in-area?	(Position \rightarrow Boolean)	Zal #t of #f teruggeven of deze positie in de omgeving zit van de meegegeven positie.
close-enough?	(Position \rightarrow Boolean)	Zal #t of #f teruggeven of deze positie dicht in de buurt is van de meegegeven positie.
outside-playarea?	(Number, Number \rightarrow Boolean)	Zal #t of #f teruggeven of deze positie zich buiten het speel veld bevindt.
display-position	($\emptyset \rightarrow \emptyset$)	Gaat de waarden van de ADT op een ordelijke manier afprinten.
set-distance-num!	(Number $\rightarrow \emptyset$)	Zet de afstand gelijk aan de meegegeven waarde.
distance-num	Number	Geeft de distance-num terug van de Position ADT.
display-compare-position	(Position $\rightarrow \emptyset$)	Gaat op een ordelijke manier de meegegeven positie vergelijken met de Position ADT.

overshooting?	(Position, Number, Number -> Boolean)	Gaat een boolean teruggeven of met de meegegeven waarde bij de volgende stap er over de volgende positie gestapt gaat worden.
calculate-distance	(Position, Number)	Berekent de afstand tussen de Position ADT en de meegegeven positie.

f) Monster ADT

Het Monster ADT creëert een monster wat de vijand is in het spel. Elke monster heeft een positie die opgeroepen en veranderd kan worden. Het plaatsen en verwijderen van monsters zal gebeuren door het Tower ADT en het Game ADT.

Naam	Signatuur	Beschrijving
make-monster	(Position -> Monster)	Constructor
get-position	Position	Geeft de positie terug van het monster.
set-position	(Position -> \emptyset)	Zet de positie gelijk aan de meegegeven positie.
get-tile	Tile	Zal de tile teruggeven van dit monster.
set-increment!	(Number, Number -> \emptyset)	Zal de x en y coördinaten van het increment aanpassen naar de meegegeven nummers.
endpoint?	Boolean	Zal #t of #f teruggeven of het monster het einde van het pad heeft bereikt.
entity?	'monster	Zal 'monster teruggeven als identificatie.
set-next-position!	(\emptyset -> \emptyset)	Zal de nieuwe positie bereken voor het monster.
get-health	Number	Geeft de health terug van het monster.
hit!	(\emptyset -> \emptyset)	Zal een leven van het monster aftrekken.
set-angle!	(Number -> \emptyset)	Zet de angle gelijk aan de meegegeven angle.
get-angle	Number	Geeft de angle terug.
get-speed	Number	Geeft de speed terug.
set-last-path-position!	(Position -> \emptyset)	Zet de laatste bezochten path positie.
get-last-path-position	Position	Geeft de laatst bezochte path positie terug.
get-random-event	Procedure	Geeft een procedure terug die het random event bevat van het monster.
on-death	Procedure	Geeft een procedure terug over wat er moet gebeuren als een monster sterft.
set-health!	(Number -> \emptyset)	Zet de health van het monster gelijk aan de meegegeven waarde.

get-type	Type	Geeft het type terug van het monster.
set-speed!	(Number -> ∅)	Zet de snelheid gelijk aan de meegegeven waarde.
get-damage	Number	Geeft de damage terug van het monster.
set-infection!	(Number -> ∅)	Zet de infection gelijk aan de meegegeven waarde.
get-infection	Number or #f	Geeft #f terug of de cooldown time van de infection.
get-default-speed	Number	Geeft de default snelheid terug van het monster.
get-points	Number	Geeft de hoeveelheid punten terug die de speler krijgt voor het verslagen van het monster.
add-passed-obstacle!	(Obstacle -> ∅)	Voegt het gepasseerde obstacle toe aan de lijst.
get-passed-obstacles	List	Geeft de lijst terug van gepasseerde obstacles.

g) Draw ADT

Het Draw ADT is verantwoordelijk voor alles wat te maken heeft met de grafische layout van het spel. Het Draw ADT kan eender welk object met een meegegeven positie teken op het scherm. Het zal ook zorgen voor het tekenen van de spelwereld.

Naam	Signatuur	Beschrijving
make-draw	Draw	Constructor
draw!	(Object, Position -> ∅)	Zal het meegegeven object tekenen op de meegegeven locatie.
draw-world!	(∅ -> ∅)	Zal de wereld (speelveld) tekenen op het scherm.
reposition!	(Tile, Position -> ∅)	Zal de meegegeven tile tekenen op het scherm op de meegegeven positie.
get-window	Window	Geeft de window terug.
entity-layer	Layer	Geeft de entity layer terug.
tower-layer	Layer	Geeft de tower layer terug.
projectile-layer	Layer	Geeft de projectile layer terug.
background-layer	Layer	Geeft de background layer terug.
draw-start-text	(∅ -> ∅)	Procedure die de start instructie tekst weergeeft.
get-start-text	String	Geeft de start tekst terug.
get-text-layer	Layer	Geeft de start tekst layer terug.
draw-game-status-text	(∅ -> ∅)	Hertekent alle informatie op het beeld.
remove-start-text!	(∅ -> ∅)	Verwijderd de start tekst van het scherm.
remove-all-text!	(∅ -> ∅)	Verwijderd alle tekst van het scherm.

remove-previous-text!	($\emptyset \rightarrow \emptyset$)	Verwijdert alle vorige tekst van het scherm.
draw-game-over-screen!	($\emptyset \rightarrow \emptyset$)	Tekent de game over scherm.
remove-game-over-screen!	($\emptyset \rightarrow \emptyset$)	Verwijdert de game over screen van het scherm.
get-power-up-layer	Layer	Geeft de power-up layer terug.
add-portal-opacity!	($\emptyset \rightarrow \emptyset$)	Voegt een 50% opacity toe aan de portal dummy in het menu.
remove-portal-opacity!	($\emptyset \rightarrow \emptyset$)	Verwijdert de 50% opacity van de dummy portal in het menu.
add-bomb-opacity!	($\emptyset \rightarrow \emptyset$)	Voegt een 50% opacity toe aan de bom dummy in het menu.
remove-bomb-opacity!	($\emptyset \rightarrow \emptyset$)	Verwijdert de 50% opacity van de dummy bom in het menu.
reset!	(Boolean $\rightarrow \emptyset$)	Gaat het hele ADT resetten en kijken of het de eerste keer is of niet.
draw-wave-text!	($\emptyset \rightarrow \emptyset$)	Tekent de wave tekst met de waarde van de wave.
remove-wave-text!	($\emptyset \rightarrow \emptyset$)	Verwijdert de wave tekst van het scherm.
update-wave-count!	($\emptyset \rightarrow \emptyset$)	Hertekent de wave tekst met de nieuwe wave waarde.
remove-wave-count!	($\emptyset \rightarrow \emptyset$)	Verwijdert de wave count van het scherm.

h) Projectile ADT

Het projectiel ADT is het ADT dat voor het verslagen van monsters zal zorgen. Als een toren de mogelijkheid heeft om een monster aan te vallen zal deze een projectiel richting het monster sturen en deze raken. Als een monster geraakt wordt zal deze schade krijgen aan de hand van het type projectiel.

Naam	Signatuur	Beschrijving
make-projectile	(position, target, environment \rightarrow projectile)	Constructor
get-tile	Tile	Geeft de tile terug van het projectiel.
get-position	Position	Geeft de positie terug van het projectiel.
move!	($\emptyset \rightarrow \emptyset$)	Procedure die eerst de richting zal berekenen en hierna een stap naar voor zal zetten in die berekende richting.
remove-projectile	($\emptyset \rightarrow \emptyset$)	Verwijdert het projectiel van het scherm en van de toren.

get-behaviour	Procedure	Geeft het gedrach terug van het projectiel.
minus-time!	(Number -> \emptyset)	Trekt de meegegeven tijd af van zijn timer.
get-timer	Number	Geeft de timer terug.
obstacle?	Boolean	Geeft terug of het projectiel een obstacle is of niet.
get-move	Boolean	Geeft terug of het projectiel mag bewegen of niet.
set-move!	(Boolean -> \emptyset)	Zet de move? gelijk aan de meegegeven waarde.
get-type	Type	Geeft het type terug van het projectiel.
set-tile!	(Tile -> \emptyset)	Zet de tile gelijk aan de meegegeven waarde.
entity?	'power-up	Geeft het entity type terug.
set-angle!	(Number -> \emptyset)	Zet de angle gelijk aan de meegegeven waarde.
exploded?	Boolean	Geeft terug of (als het projectile van het type 'bomb is) het projectiel al geexplodeerd is.
set-projectile?	(Boolean -> \emptyset)	Zet projectile? Gelijk aan de meegegeven waarden.
projectile?	Boolean	Geeft terug of het projectiel ook een projectiel is. (De bom is een projectiel maar wordt ook gebruikt als power up)

i) Wave ADT

Het wave ADT zal voor het toevoegen van monsters in het spel zorgen. Elke wave zal er een lijst opgevuld worden met monsters die gebruikt zullen worden in die ronde. Deze lijst zal dan meegegeven worden aan het environment ADT die voor het tekenen van de monsters zal zorgen.

Naam	Signatuur	Beschrijving
make-wave	(Environment -> Wave)	Constructor
next-monster!	(\emptyset -> \emptyset)	Zal het volgende monster in de lijst in toevoegen aan het spel.
load-wave!	(\emptyset -> \emptyset)	Zal voor de gegeven wave de lijst vullen met de monsters voor die wave.
get-wave	Wave	Geeft de wave terug waar men op dat moment zit.
set-wave!	(val -> \emptyset)	Zal de wave gelijkzetten aan de meegegeven waarde.
set-wave-list!	(val -> \emptyset)	Zal de wave list gelijkzetten aan de meegegeven waarde.
get-wave-list	Lijst	Geeft de lijst terug met alle monsters die gebruikt zullen worden in die wave.

j) Player

Het Player ADT zal de punten, levens, en power-ups van de speler bijhouden. Als de speler geen levens meer heeft is het spel gedaan.

Naam	Signatuur	Beschrijving
make-player	(Number -> Player)	Constructor
get-health	Number	Geeft de levens van de speler terug.
set-health!	(Number -> ∅)	Zal de waarde van de levens van de speler aan passen naar de meegegeven waarde.
hit!	(Number -> ∅)	Zal set-health! Oproepen met de meegegeven waarde en de levens ervan aftrekken.
get-points	Number	Geeft de punten terug van de speler.
set-points!	(Number -> ∅)	Zal de punten van de speler aanpassen naar de meegegeven waarde.

k) Power-up

De power-ups zijn de wapens die de speler zelf kan gebruiken om monsters te verslaan. Elke power-up heeft een cooldown. Je hebt de bommenwerper, waarbij de speler een bom op het pad kan gooien en de monsters die in het bereik van de bom zijn zullen schade krijgen. Je hebt de teleporter, waarbij als er een monster over de geplaatste teleport loopt, deze terug gestuurd zal worden naar het begin van het pad.

Naam	Signatuur	Beschrijving
make-power-up	(String, Location -> Power-up)	Constructor
get-cooldown	Number	Geeft de cooldown terug van de power-up.
set-cooldown!	(Number -> ∅)	Zal de cooldown aanpassen naar de meegegeven waarden.
get-type	String	Geeft terug wat voor een power-up het is.
calculate-radius!	(∅ -> ∅)	Zal de schade omgeving berekenen van de bom power-up of de teleport
get-damage	Number	Zal de damage teruggeven van de bom.
set-damage!	(Number -> ∅)	Zet de waarde van de damage van de bom gelijk aan de meegegeven waarde.
teleport!	(Monster -> ∅)	Zal het meegegeven monster terug teleporteren naar het begin van het pad.
get-price	Number	Geeft de prijs terug van de power-up.
set-price!	(Number -> ∅)	Zal de prijs van de power-up veranderen naar de meegegeven waarde.

l) Menu

Het menu is de plaats waar de speler torens en power-ups kan kopen of activeren.

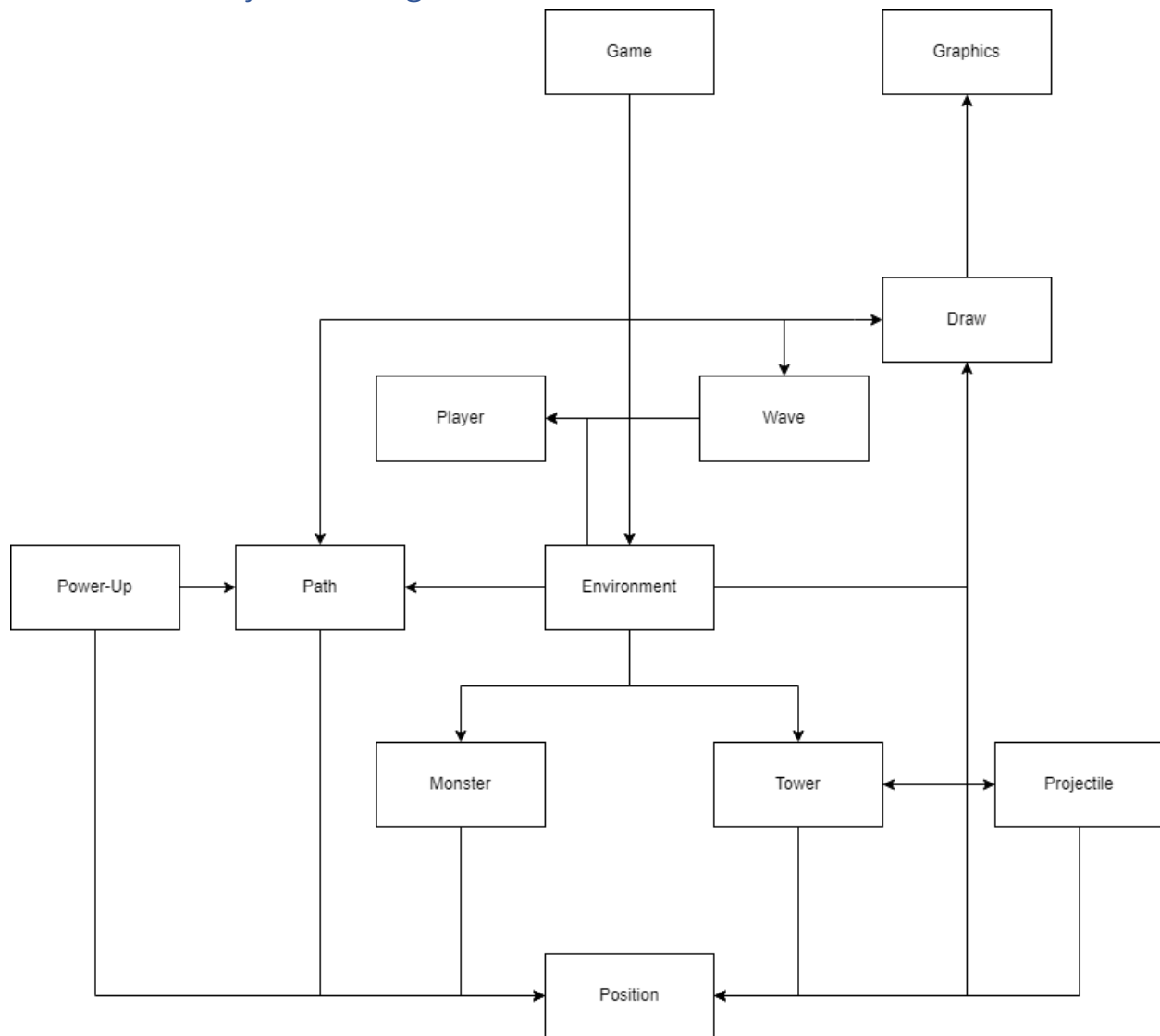
Naam	Signatuur	Beschrijving
Make-menu	Menu	Constructor
get-towers	List	Geeft de lijst terug van alle torens die gebruikt kunnen worden.
get-power-ups	List	Geeft de lijst terug van alle power-ups die gebruikt kunnen worden.

m) Procedures

Er wordt ook gebruik gemaakt van een procedures file, waar allemaal algemene procedures in zitten om het project ordelijker te houden.

Naam	Signatuur	Beschrijving
Make-menu	Menu	Constructor
object-eq-pos-obstacles?	(Object, List -> Boolean)	Geeft een boolean terug die zegt of het meegegeven object gelijk is aan een obstacle uit de meegegeven lijst.
print-el-in-list	($\emptyset \rightarrow \emptyset$)	Print op een orderlijke manier alle element uit van de meegegeven lijst.
remove-el-from-list	(Element, List -> List)	Geeft de lijst terug zonder het meegegeven element als deze bestaat in de lijst.
add-element-to-list	(Element, List -> List)	Geeft de lijst terug met het meegegeven element toegevoegd aan de lijst.
pick-random-from-list	(List (optioneel Element) -> Element)	Geeft een random element terug van de meegegeven lijst buiten het meegegeven element (Als deze is meegegeven).
random-pos-between-points	(Position, Position -> Position)	Geeft een random positie terug dat tussen de twee meegegeven posities ligt.
create-target-pos	(Monster -> Position)	Creëert een positie van het meegegeven monster.
set-tile-position!	(Tile, Position -> Tile)	Veranderd de positie van de meegegeven tile gelijk aan de meegegeven positie en geeft de tile dan terug.

3. Afhankelijkheidsdiagram



Het afhankelijkheidsdiagram toont het objectgeoriënteerd ontwerp van alle ADTs in het spel. In het spel zijn er 11 verschillende ADTs. De pijl tussen 2 ADTs beschrijft de onderlinge relatie. De relaties worden hieronder beschreven:

- Het Game ADT is afhankelijk van het Environment ADT en het Wave ADT, omdat deze procedures oproept die aanwezig zijn in het Environment ADT en het Wave ADT.
- Het Environment ADT is afhankelijk van het Draw, Path, Tower, Monsteren Player ADT, omdat het Environment ADT alle logica en informatie bevat van het spel en zal daarom alle elementen van het spel moeten aanroepen en de gegevens van de speler.
- Het Draw ADT is afhankelijk van de “Graphics.rkt” library en het Position ADT om het tekenen op het scherm mogelijk te maken. Het Draw ADT heeft ook een positie nodig om te weten waar een object op het scherm getekend moet worden.
- Het Path ADT is afhankelijk van het Position ADT, omdat het Path ADT alle belangrijke posities van het pad bezit.

- Het Tower ADT is afhankelijk van het Position en Projectile ADT, omdat elke toren dat geplaatst zal worden in het spel een positie moet krijgen en een projectiel moet kunnen afvuren.
- Het Monster ADT is afhankelijk van het Position ADT, omdat elk monster zich moet kunnen verplaatsen over het pad.
- Het Wave ADT is afhankelijk van het Environment ADT voor het toevoegen van de monsters aan het environment.
- Het Player ADT is afhankelijk van niets.
- Het Power-up ADT is afhankelijk van het Path en Position ADT voor zich te kunnen positioneren op het pad.
- Het Projectile ADT is afhankelijk van het Tower en Position ADT om te weten van welke toren hij komt en hoe hij zich moet kunnen positioneren.
- Het Position ADT is afhankelijk van niets.

4. Tijdschema

Week	Actie
30	Implementatie Player ADT (1 dag)
31	Implementatie Power-up ADT (7 dagen)
32	Implementatie Menu ADT + aanpassing Environment ADT (4 dagen)
33	Aanpassing + Implementatie Game, Path ADT (4 dagen)
34	Aanpassing + implementatie Monster, Tower ADT (3 dagen)
35	Aanpassing + implementatie Wave ADT en Projectile ADT (6 dagen)
36	Alles opschonen en bug fixing (7 dagen)
/	Projectverdediging

5. Instructies van het spel

- Voor het opstarten van het spel zul je de code moeten uitvoeren in het “Main.rkt” bestand:
- De tekst “Press: s to start...” Zal nu verschijnen in een window. Volg de instructies om het spel te beginnen:

- Nu zal de wereld verschijnen, maar gaat er voor de rest niets gebeuren:



- Op het scherm zie je nu je levens, punten een menu en aan welke wave je nu zit.
- Nu heb je oneindig veel tijd om je voor te bereiden op de eerste wave. Je begint met 200 punten en kan hiermee torens kopen in het menu.
- Bij het drukken op een toren wordt deze geselecteerd en kan je deze op een vrije plaats plaatsen in de wereld en pas dan worden de punten afgetrokken.
- Het menu bestaat uit:
 - Towers:



- -> De kostprijs van deze toren bedraagt 100 punten. Een standaardtoren die op een vaste frequentie een steen afvuurt naar een monstertje. Een monstertje verliest 1 leven bij aanraking van de steen, waarbij de steen onmiddellijk verdwijnt.



- -> De kostprijs van deze toren bedraagt 100 punten. Een toren die op een lage frequentie een net werpt naar de monstertjes. Monstertjes die het net aanraken zullen voor een bepaalde tijd trager bewegen. Het net doet geen schade en stopt onmiddellijk met bewegen vanaf zodra het een monstertje aangeraakt heeft om vervolgens enkele seconden later te verdwijnen.



- -> De kostprijs van deze toren bedraagt 150 punten. Een toren die op een vaste frequentie acht kogels afvuurt (twee horizontaal, twee verticaal en vier diagonaal) wanneer er een monstertje in de buurt is. Monstertjes

verliezen 1 leven bij aanraking van een kogel. De kogels verdwijnen onmiddellijk na de aanraking.



- -> De kostprijs van deze toren bedraagt 200 punten. Een toren die aan een zeer lage frequentie een bom werpt richting het dichtstbijzijnde pad (als er meerdere monstertjes in de buurt zijn). De bom explodeert na enkele tellen en alle monstertjes in de buurt van de bom verliezen 3 levens (de meeste monstertjes zullen dus onmiddellijk sterven).

- Power-ups:

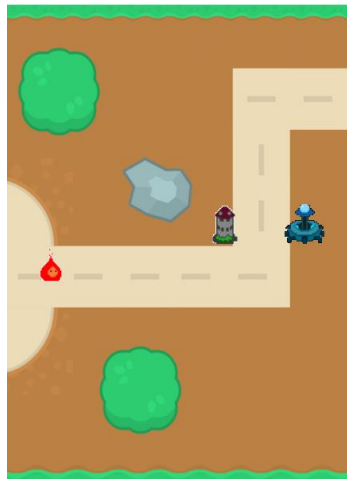


- ->Teleporter. Wanneer deze power-up geactiveerd wordt verschijnen er twee teleporters op het pad op twee willekeurige posities (maar niet naast elkaar). De teleporter die het dichtste bij het einde van het pad staat teleporteert monstertjes naar de andere teleporter. Elke teleporter heeft een verschillende kleur. Deze power-up heeft een cooldown van 10 seconden.



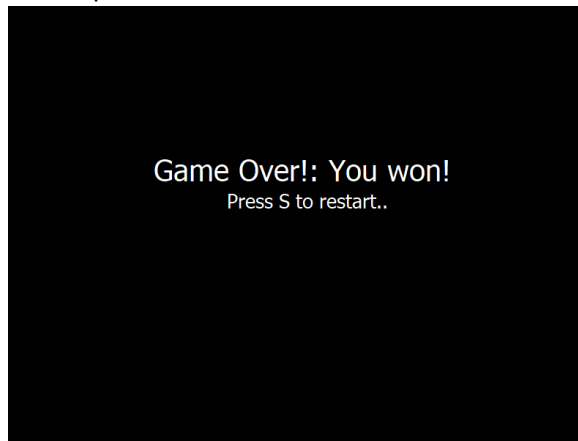
- ->Bommenwerper. Wanneer deze power-up geactiveerd wordt vallen er enkele bommen op willekeurige plaatsen in de spelwereld. De monstertjes die door de bommen geraakt worden verliezen willekeurig 1 of 2 levens. De bommen moeten kort zichtbaar blijven voor ze exploderen. Deze power-up heeft een cooldown van 10 seconden.

- Als de voorbereiding van de wave klaar is kan de speler de wave starten door, als aangegeven, op de toets 'w' te drukken:
- Nu zullen de monsters van de wave om de beurt in de wereld geplaatst worden:

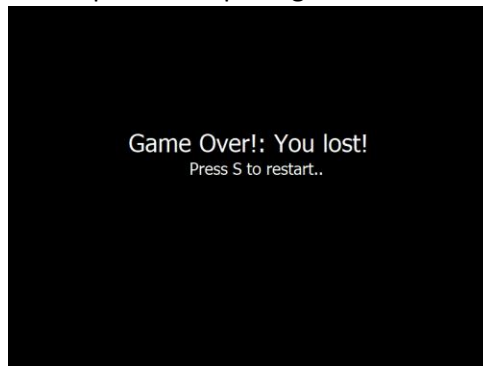


- Als een monster het einde van het pad bereikt zullen er levens worden afgetrokken van de speler afhankelijk van het monster:

- Het spel eindigt als, of alle monsters en alle waves zijn verslagen, of als de spelers geen levens meer heeft:
 - Einde spel door alle monsters en waves te verslagen:



- Einde spel als de speler geen levens meer heeft:



- Je kunt het spel herstarten door op de knop 'S' te drukken op het toetsenbord.

6. Fouten en onbrekende functionaliteiten

- Een bom toren gaat niet een bom naar het dichtstbijzijnde pad sturen maar naar het eerste monster dat in zijn omgeving komt.
- Bij het verslagen van een grijs monster worden er wel daadwerkelijk 2 nieuwe rode monsters aan het begin van het pad geplaatst, maar deze overlappen elkaar en is moeilijk te zien dat het er 2 zijn.
- Torens kunnen geplaatst worden onder het menu waar geen icoon staat.