

# 2015

**Hogeschool van Arnhem en Nijmegen**

Semester: OOSE  
Course: OOAD  
Docent Ben Nieuwhof

Els den Engelse 480913  
Walter Jansen 529530  
Ben Rens 533123

## **CASE STUDY 2: KISS OF DEATH**

# Inhoudsopgave

1	Inleiding.....	3
2	Requirements.....	4
2.1	Functional.....	4
2.2	Usability.....	5
2.3	Reliability.....	5
2.4	Performance.....	5
2.5	Supportability.....	5
4	Use Case diagram.....	6
5	Use Cases.....	7
5.1	Use Case: Bewegen.....	7
5.1.1	Use Case: Power-up pakken.....	8
5.1.2	Use Case: Attribuut pakken.....	9
5.3	UC. Kussen.....	10
5.4	UC. Aanvallen en Verdedigen.....	11
5.4.1	Use Case: Aanvallen.....	11
5.4.2	Use Case: Verdedigen.....	12
5.5	Use Case: Aanmelden bij het spel.....	13
5.6	Use Case: Spel aanmaken.....	13
5.7	Use Case: Spel verlaten.....	13
5.8	Use Case: Power-up plaatsen.....	13
5.9	Use Case: Power-up verwijderen.....	13
6	Domein model.....	14
7	Activity Diagrammen.....	15
7.1	Boodschap versturen.....	15
7.2	Bewegen.....	16
7.3	Aanvallen.....	17
8	System Sequence Diagrammen.....	18
8.1	Boodschap versturen.....	18
8.2	Bewegen.....	18
8.3	Kussen.....	19
8.4	Aanvallen.....	19
8.5	Verdedigen.....	20
9	Sequence Diagrammen.....	21
9.1	Boodschap sturen.....	21
9.2	Bewegen.....	22

9.3	Kussen.....	23
9.4	Aanvallen .....	25
10	State Machine Diagram.....	26
11	Design Class Diagram .....	27
12	Architectuurmodel .....	28

# 1 Inleiding

Welkom bij de kiss of death. Dit is een spel waarbij het draait om het monster te vinden, die de gedaante heeft aangenomen van een strijder, een elf of een dwerg. Samen met je team moet je proberen dit monster te vinden, maar hij kan natuurlijk ook bij jou in het team zitten. Val wezens aan van de tegenpartij of van je eigen team. Weet je iemand te verslaan en is dit het monster, dan verandert hij in zijn ware gedaante. De enige manier om het monster te verslaan, is door hem te kussen. Lukt dit, dan wint je team. Weet het monster iedereen te verslaan, dan wint het monster. Kus je iemand van de tegenpartij en is het geen monster? Dan verander je in een kikker en moet je maar hopen dat een medespeler jou kust, want als kikker zijnde kun je niks beginnen.

In dit document is beschreven wat het spel bevat, hoe het spel werkt en hoe het er in werkende code zou moeten opleveren. Door gebruik te hebben gemaakt van verschillende UML diagrammen hebben wij het systeem zo goed mogelijk ontworpen.

## 2 Requirements

### 2.1 Functional

- F1. Het spel bevat een aantal aan elkaar grenzende kamers
- F2. Het spel kan pas gestart worden als er 9 spelers zijn aangemeld.
- F3. Per team (Elf, Strijder en Dwerg) worden 3 spelers ingedeeld.
- F4. Spelers van een team kunnen elkaar en anderen in dezelfde kamer boodschappen sturen.
- F5. Eén van de 9 spelers is een monster.
- F6. Als het monster het spel verlaat, is het spel afgelopen en wint het team met de meeste levenspunten
- F7. Als een andere speler het spel verlaat, gaat het spel verder zonder die speler.
- F8. Spelers kunnen zich verplaatsen naar een aangrenzend vakje, mits hun energie groter is dan 10.
- F9. Spelers mogen ook sneller bewegen, maar dat kost meer energie.
- F10. Spelers kunnen alleen attributen pakken die past bij het spelerstype.
- F11. Als een wezen zonder een attribuut op een vakje wat een attribuut bevat komt, pakt hij dit attribuut op.
- F12. Het meedragen van een attribuut kost extra energie.
- F13. Spelers kunnen een attribuut neerleggen.
- F14. Als een wezen op een vakje komt met een bezemsteel, maakt hij automatisch er automatisch gebruik van.
- F15. Na het verplaatsen met een bezemsteel kan het wezen <<init>> seconden geen gebruik maken van de bezemsteel.
- F16. Als een speler een wezen kust wat geen monster of kikker is, kost dat de speler de helft van zijn levenspunten.
- F17. Wanneer een speler wordt gekust door een speler van een ander type, verandert het in een kikker.
- F18. Een kikker kan weer een speler worden door gekust te worden door een wezen van hetzelfde type.
- F19. Als het monster wordt gekust, is het verslagen en heeft het team van de kusser gewonnen.
- F20. Als de dekmantel van het monster is verslagen, wisselt het monster van gedaante en is het voor alle spelers te zien wie het monster is.
- F21. Spelers kunnen elkaar alleen aanvallen als de ander op een aangrenzend vakje staat.
- F22. Aanvallen kost <<init>> energie.
- F23. Een aanval slaagt of mislukt.
- F24. Als er niet verdedigd wordt, is de kans 80% dat de aanval lukt als de aanvaller een attribuut heeft, en anders 70%.
- F25. Als er verdedigd wordt, daalt de kans naar 50% als beiden een attribuut hebben of beiden geen attribuut hebben; als alleen de aanvaller een attribuut heeft is de kans 60%, en als alleen de verdediger een attribuut heeft is de kans 40%.
- F26. Bedreigde spelers kunnen zich verdedigen tegen een aanval.
- F27. Verdedigen kost <<init>> energie.
- F28. Het spel wordt gespeeld in 2d.
- F29. Bij 0 levenspunten is het wezen dood.

## **2.2 Usability**

- U1.** Het spel moet gemakkelijk bespeelbaar zijn.

## **2.3 Reliability**

- R1.** De server handelt alle acties af.

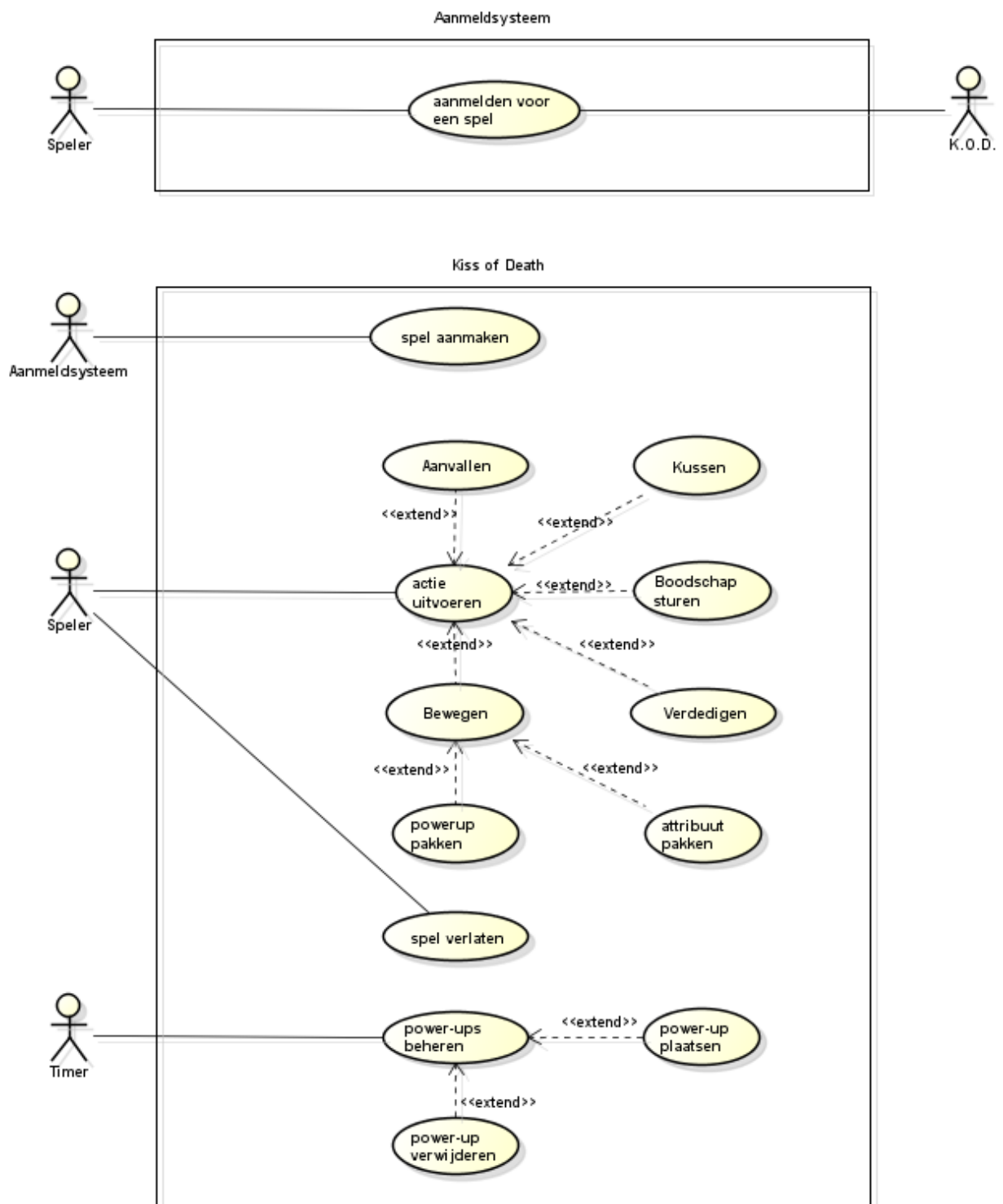
## **2.4 Performance**

- P1.** Het spel kan altijd gestopt worden.
- P2.** Acties worden gelijk uitgevoerd.
- P3.** De gebruikers horen continue updates te krijgen.

## **2.5 Supportability**

- S1.** Een cliëntvenster behoort bij één specifieke speler.
- S2.** Het spel draait op een server

## 4 Use Case diagram



## 5 Use Cases

### 5.1 Use Case: Bewegen

<b>Primary actor:</b> Speler	
Stakeholders: Server en andere spelers	
<b>Brief description:</b> De speler verplaatst zich naar een andere positie in het speelveld.	
<b>Preconditions:</b> De speler moet over meer dan 10 energiepunten beschikken	
<b>Postconditions:</b> De speler is verplaatst naar het gewenste vakje.	
<b>Basic Flow</b>	
Actor Action	System Response
1. Speler geeft aan te willen verplaatsen.	
	2. Controleer of de gevraagde plek al bezet is.
	3. Verplaats de speler naar de aangewezen plek.
<b>Alternatieve Flow: Speler wil sneller bewegen</b>	
1b. Speler geeft aan te willen verplaatsen met dubbele snelheid	
	2b. Controleer of de gevraagde plekken al bezet zijn.
	3b. Verplaats de speler naar de aangewezen plek.
<b>Alternatieve Flow: Er bevindt zich een power-up op de aangewezen plek</b>	
	4c. Systeem constateert dat er een power-up op de plek ligt.
	5c. Use case wordt vervolgt in power-up oppakken.
<b>Alternatieve Flow: Er bevindt zich een attribuut op de aangewezen plek</b>	
	4d. Systeem constateert dat er een attribuut op de plek ligt.
	5d. Use case wordt vervolgt in attribuut oppakken.
<b>Alternatieve Flow: Het gekozen vakje is bezet</b>	
	3e. Systeem constateert dat de gekozen plek bezet.
	4e. De actie wordt geannuleerd.



### 5.1.1 Use Case: Power-up pakken

<b>Primary actor:</b> Positietrigger (onder voorbehoud)	
<b>Stakeholders:</b> Spelers	
<b>Brief description:</b> Wanneer de speler zich beweegt naar een plek waar zich een power-up bevindt, dan wordt de positietrigger geactiveerd. Wanneer het om een vitamine gaat, wordt er energie aan de desbetreffende speler toegevoegd. Als het om een bezem gaat krijgt de speler de mogelijkheid om zich snel naar een andere positie te begeven.	
<b>Preconditions:</b> Power-up ligt op een vakje.	
<b>Postconditions:</b> Power-up is succesvol uitgevoerd op de speler.	
<b>Basic Flow</b>	
Actor Action	System Response
1. Geef de positie van de speler.	
	2. Zoek type power-up bij de positie.
	3. Het type is een vitamine.
	4. Het systeem geeft speler <<init>> energie en verwijderd de vitamine uit het systeem.
<b>Alternative Flow: De power-up is een bezem</b>	
	3b. Het type is een bezem.
	4b. Het systeem laat gebruiker een positie kiezen.
5b. Selecteer een nieuwe positie	
	6b. Verplaats de speler naar gekozen positie.
	7b. Plaats de bezem op de zelfde positie als de speler.
	8b. Zet de cooldown van de speler op 15 seconde. <<init>>

### 5.1.2 Use Case: Attribuut pakken

<b>Primary actor:</b> Positietrigger (onder voorbehoud)	
<b>Stakeholders:</b> Spelers	
<b>Brief description:</b> Wanneer de speler zich beweegt naar een plek waar zich een attribuut bevind, dan wordt de positietrigger geactiveerd. Er wordt gekeken of het attribuut dat op de grond ligt, door het spelkarakter opgepakt kan worden.	
<b>Preconditions:</b> Het spelkarakter heeft geen attribuut en het attribuut moet overeenkomen met het karakter.	
<b>Postconditions:</b> Attribuut is bij de speler gevoegd en van de grond verwijderd.	
<b>Basic Flow</b>	
Actor Action	System Response
1. Geef de positie van de speler.	
	2. Zoek type attribuut bij de positie.
	3. Controleer of het type attribuut pas bij het type speler
	4. Het attribuut wordt toegevoegd aan de speler en van de grond verwijderd.
<b>Alternative Flow : Attribuut hoort niet bij speler</b>	
	4b. Het attribuut blijft liggen.

### 5.3 UC. Kussen

<b>Primary actor:</b> Speler	
Stakeholders: K.O.D, medespelers	
<b>Brief description:</b> Een speler voert de actie kussen naar een richting, hij kust een medespeler van hetzelfde type wezen hierdoor verliest hij levenspunten <<init>>.	
<b>Preconditions:</b> De speler is geen kikker of monster en leeft nog.	
<b>Postconditions:</b> De Speler heeft een medespeler van hetzelfde type wezen gekust en verloor levenspunten.	
<b>Basic Flow</b>	
Actor Action	System Response
1. De speler geeft de actie kus uit te willen voeren, naar een bepaalde richting.	
	2a. Systeem checkt of er iemand in de kus-richting staat.
	3. Systeem checkt type wezen van medespeler
	4a. Systeem registreert dat de speler en de medespeler van hetzelfde type wezen zijn.
	5a. Systeem haalt levenspunten <<init>> van de speler af.
<b>Alternative Flow : Speler kust leegte</b>	
	2b. Systeem registreert dat er geen mede speler in de richting staat.
	3. Systeem haalt levenspunten <<init>> van de speler af.
<b>Alternative Flow : Speler kust medespeler van ander type wezen</b>	
	4b. Systeem registreert dat de speler en de medespeler verschillen qua type wezen.
	5b. Systeem verandert de rol van medespeler in een kikker
	6. Systeem haalt levenspunten <<init>> van de speler af.
<b>Alternative Flow : Speler kust kikker (success)</b>	
	4c. Systeem registreert dat de speler van hetzelfde type wezen is en de rol kikker heeft.
	5c. Systeem haalt de rol kikker bij de gekuste speler weg.
<b>Alternative Flow : Speler kust kikker (fail)</b>	
	4d. Systeem registreert dat de speler niet van hetzelfde type wezen is en de rol kikker heeft.
	5d. Systeem verandert de rol van de gekuste speler niet, hij blijft de rol kikker behouden.
<b>Alternative Flow : Speler kust monster</b>	
	4e. Systeem registreert dat de medespeler de rol monster heeft.
	5e. Systeem verklaart het team van de speler de winnaars
	6. Systeem stuurt een boodschap naar alle spelers met de uitkomst van het spel.
	7. Systeem stopt het spel.

## 5.4 UC. Aanvallen en Verdedigen

### 5.4.1 Use Case: Aanvallen

<b>Primary actor:</b> Aanvallende speler	
<b>Stakeholders:</b> K.O.D. en verdedigende speler	
<b>Brief description:</b> Een speler wil een andere speler aanvallen en doet dat door op een aanliggend vakje te gaan staan van de verdediger. Vervolgens geeft de aanvaller aan te willen aanvallen, dan heeft de verdediger nog kort de tijd om te beslissen of hij wil vluchten, verdedigen of niet gaat verdedigen. Daarbij kunnen ze allebei kiezen om een attribuut te gebruiken, mits ze deze in bezit hebben.	
<b>Preconditions:</b> Aanvaller en verdediger staan binnen een vakje van elkaar. Aanvaller heeft minimaal <<init>> energie.	
<b>Postconditions:</b> Aanvaller heeft verdediger aangevallen.	
<b>Basic Flow</b>	
Actor Action	System Response
1. Aanvaller geeft aan een speler te willen aanvallen	
	2. Systeem start vertraging
3. Aanvaller valt aan met attribuut	
	4. Systeem bepaalt de winnaar op basis van toeval en kracht.
	5. Systeem verlaagt aantal levenspunten van verliezer
<b>Alternative Flow : Mislukte aanval</b>	
	5b. Systeem verlaagt aantal levenspunten van aanvaller
<b>Alternative Flow : Verdediger vlucht</b>	
	4c. Systeem constateert dat verdediger van aanliggend vakje af is
	5c. Systeem verlaagt aantal levenspunten van aanvaller
<b>Alternative Flow : Verliezer heeft geen levenspunten meer</b>	
	5d. Systeem verlaagt aantal levenspunten van aanvaller
	6d. Systeem constateert dat verliezer geen levenspunten meer heeft
	7d. Systeem verwijdert verliezer uit het spel
<b>Alternative Flow : Aanvallen zonder attribuut, minder kans op een succesvolle aanval</b>	
3e. Aanvaller valt aan zonder attribuut	
	4e. Systeem bepaalt de winnaar op basis van toeval en kracht.
	5e. Systeem verlaagt aantal levenspunten van verliezer

### 5.4.2 Use Case: Verdedigen

<b>Primary actor:</b> Verdedigende speler	
<b>Stakeholders:</b> K.O.D. en aanvallende speler	
<b>Brief description:</b> Verdediger wordt aangevallen door een andere speler. De verdediger zal zich moeten verdedigen om een grotere kans te hebben op winnen.	
<b>Preconditions:</b> Aanvaller heeft een aanzet tot aanvallen gedaan. Verdediger heeft minimaal <<init>> energie.	
<b>Postconditions:</b> Verdediger heeft zichzelf succesvol verdedigd.	
<b>Basic Flow</b>	
Actor Action	System Response
1. Verdediger ontvangt een melding over een aanval	
2. Verdediger maakt de keuze om zich te verdedigen met een attribuut	
	3. Systeem bepaalt de winnaar op basis van toeval en kracht.
	4. Systeem verlaagt aantal levenspunten van aanvaller
<b>Alternative Flow : Verdediger wil zich niet verdedigen</b>	
2b. Verdediger geeft aan zich niet te willen verdedigen	
	3b. Systeem bepaalt de winnaar op basis van toeval en kracht.
	4b. Systeem verlaagt aantal levenspunten van verdediger
<b>Alternative Flow : Verdediger wordt al aangevallen</b>	
2c. Verdediger kan zich slechts maar tegen één aanvaller verdedigen	
	3c. Systeem bepaalt de winnaar uit beide gevechten op basis van toeval en kracht.
	4c. Systeem verlaagt aantal levenspunten van verliezer(s)
<b>Alternative Flow : Verdedigen zonder attribuut, meer kans om verslagen te worden</b>	
2d. Verdediger maakt de keuze om zich te verdedigen, maar heeft geen attribuut	
	3d. Systeem bepaalt de winnaar op basis van toeval en kracht.
	4d. Systeem verlaagt aantal levenspunten van aanvaller

## 5.5 Use Case: Aanmelden bij het spel

### **Brief description:**

Speler meldt zich via het intranet aan om het spel te spelen, vervolgens kiest de speler een nickname waar hij of zij mee speelt. Dan wacht het systeem tot er 9 spelers zijn aangemeld. Daarna wordt er een nieuw spel aangemaakt.

**Preconditions:** Speler heeft toegang tot het intranet.

**Postconditions:** Speler aangemeld voor een spel.

## 5.6 Use Case: Spel aanmaken

### **Brief description:**

Nadat er 9 spelers zijn aangemeld, wordt er een spel aangemaakt. Daarna worden alle spelers in teams ingedeeld en wordt bepaald wie het monster is. Nu wordt het spel gestart.

**Preconditions:** Er zijn 9 spelers aangemeld.

**Postconditions:** Een nieuw spel is aangemaakt, spelers zijn toegevoegd en ingedeeld in teams.

## 5.7 Use Case: Spel verlaten

### **Brief description:**

Een speler wil of moet het spel verlaten, bijvoorbeeld als het geen levenspunten meer heeft. Dan wordt de speler automatisch uit het spel verwijderd. Een monster kan ook het spel verlaten, als het is gekust door een speler.

**Preconditions:** Speler heeft geen levenspunten meer, monster is gekust of speler wil het spel verlaten.

**Postconditions:** Speler heeft het spel verlaten.

## 5.8 Use Case: Power-up plaatsen

### **Brief description:**

De timer stuurt een instructie naar het systeem om een power up te plaatsen. In deze instructie staat beschreven wat het type van de power-up is.

**Preconditions:** Het type power-up is bekend en de timer loopt af.

**Postconditions:** De power-up is op een willekeurige plaatst geplaatst.

## 5.9 Use Case: Power-up verwijderen

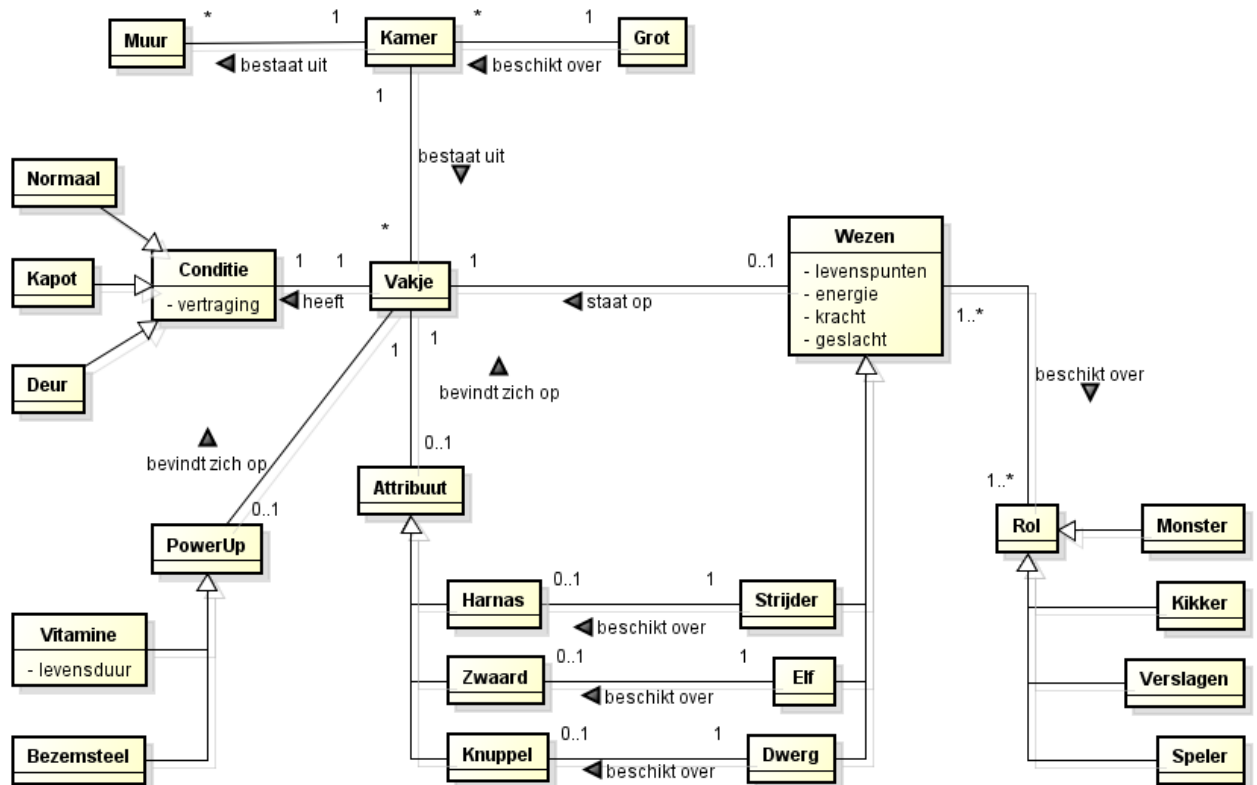
### **Brief description:**

De timer stuurt een instructie naar het systeem om een power up te verwijderen. In deze instructie staat beschreven wat het type van de power-up is.

**Preconditions:** Het type power-up is bekend, de power-up moet bestaan en de timer loopt af.

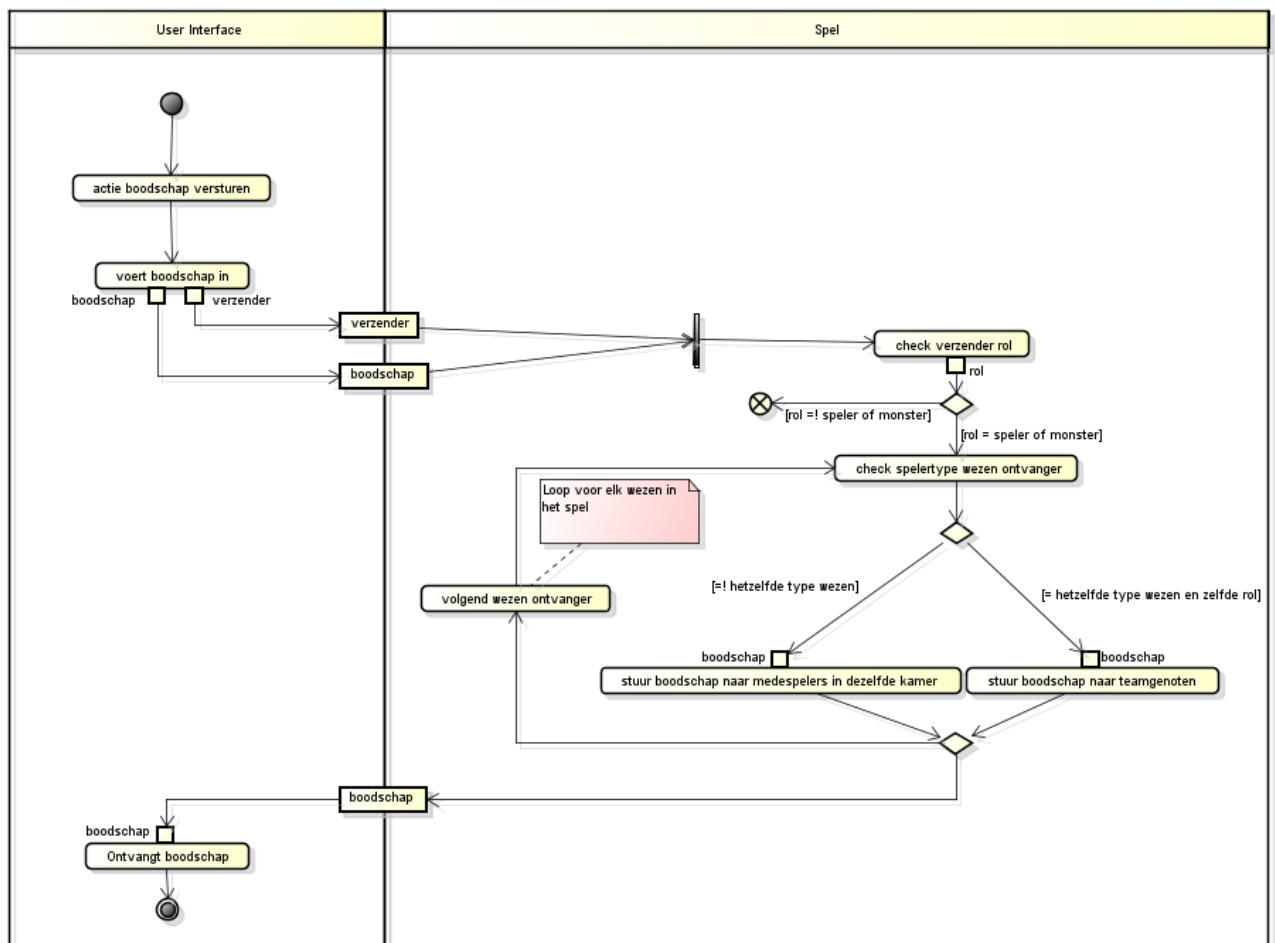
**Postconditions:** De power-up is uit het speelveld verwijderd.

## 6 Domein model



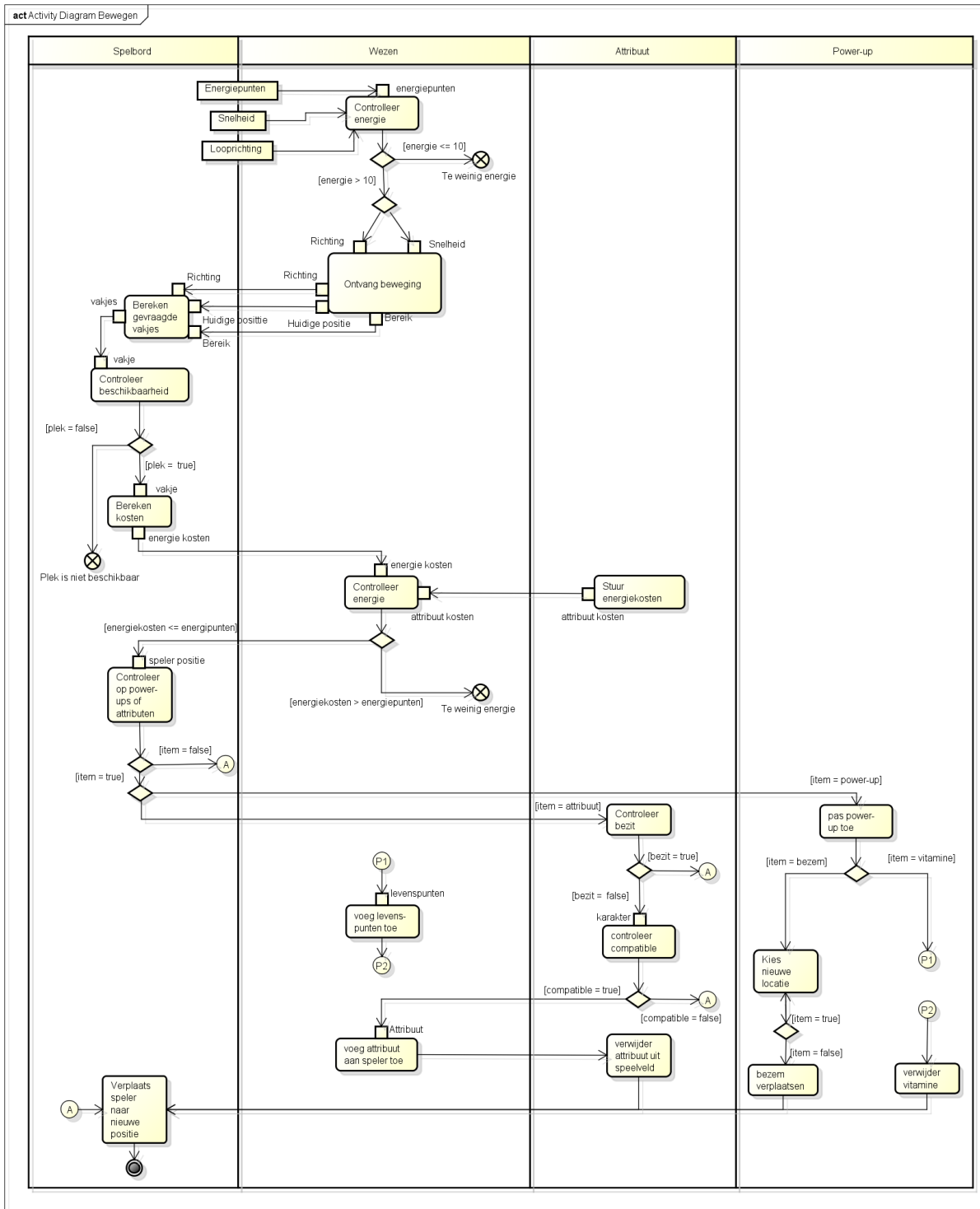
## 7 Activity Diagrammen

### 7.1 Boodschap versturen

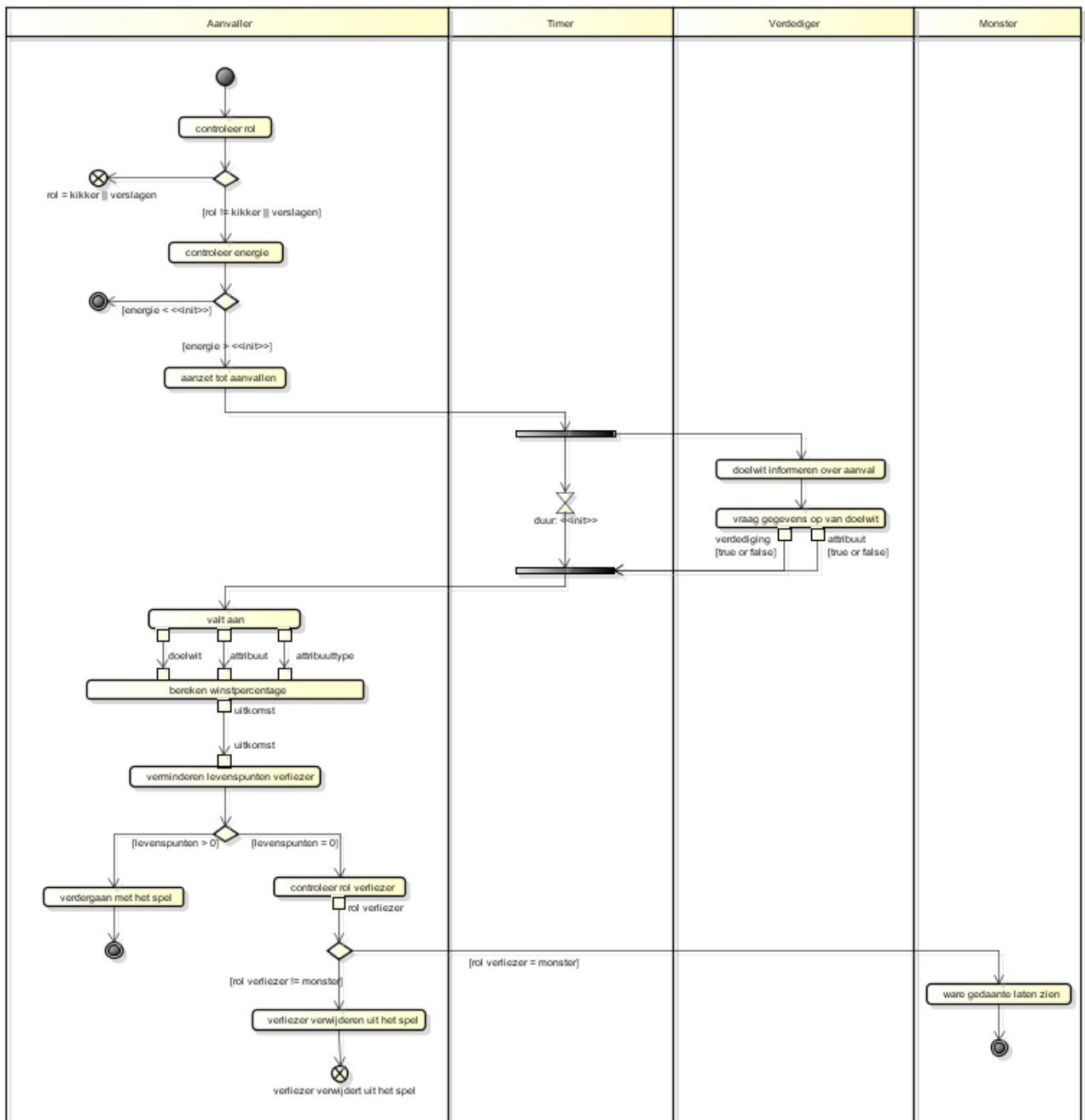




## 7.2 Bewegen

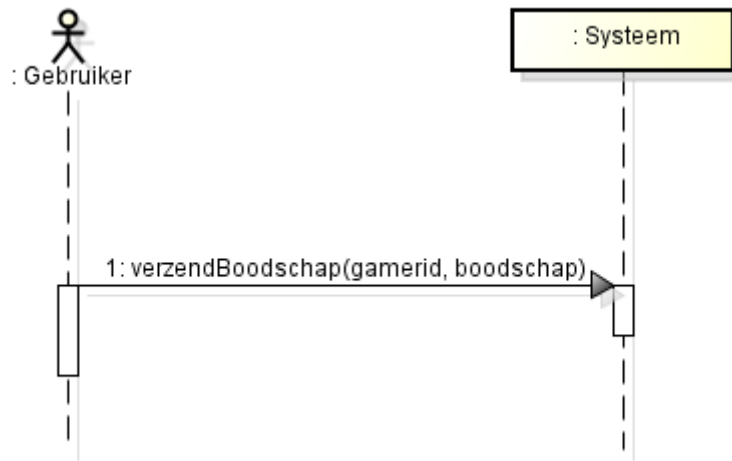


## 7.3 Aanvallen

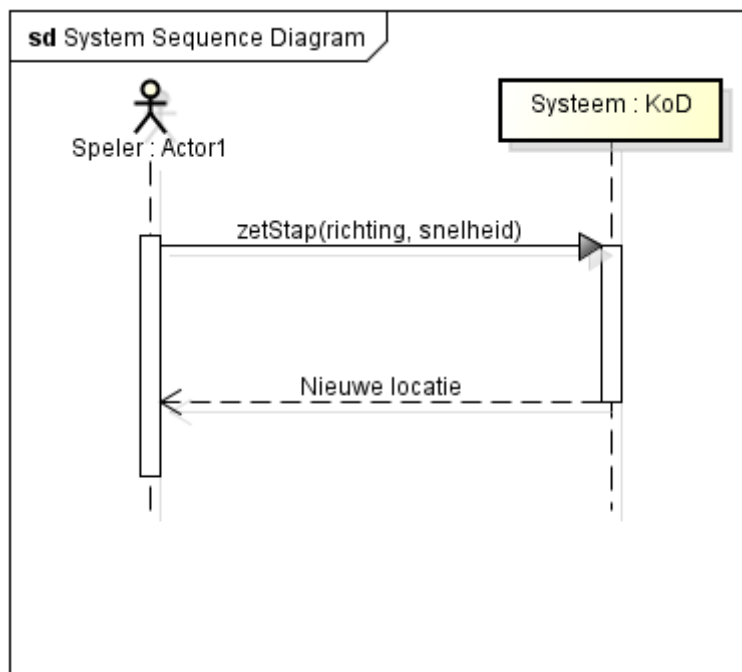


## 8 System Sequence Diagrammen

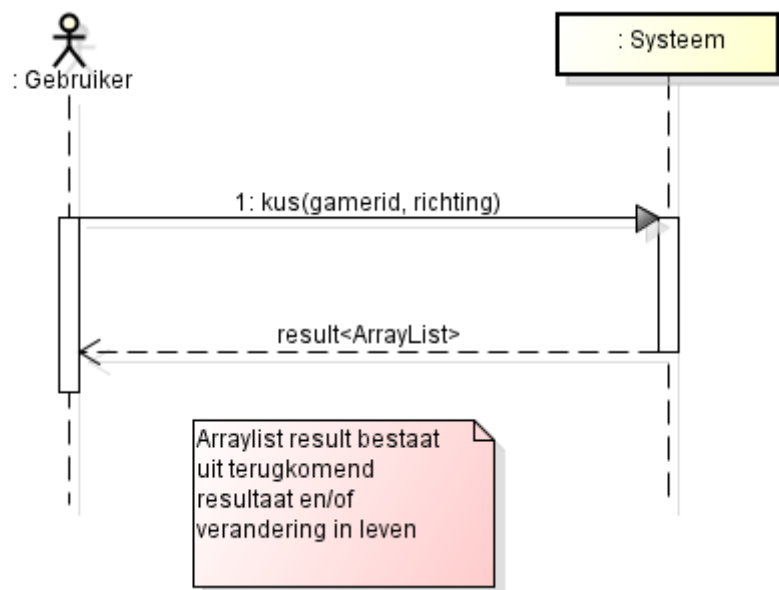
### 8.1 Boodschap versturen



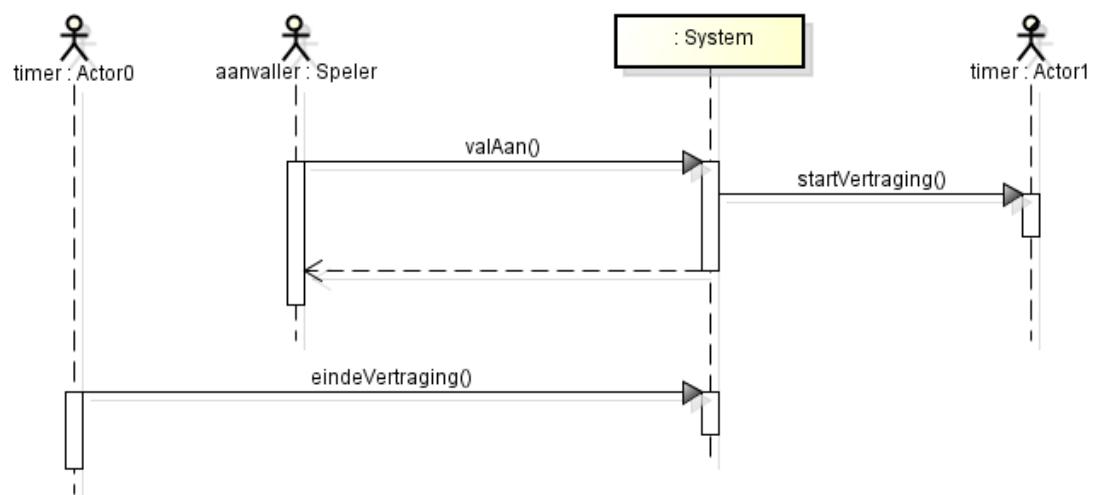
### 8.2 Bewegen



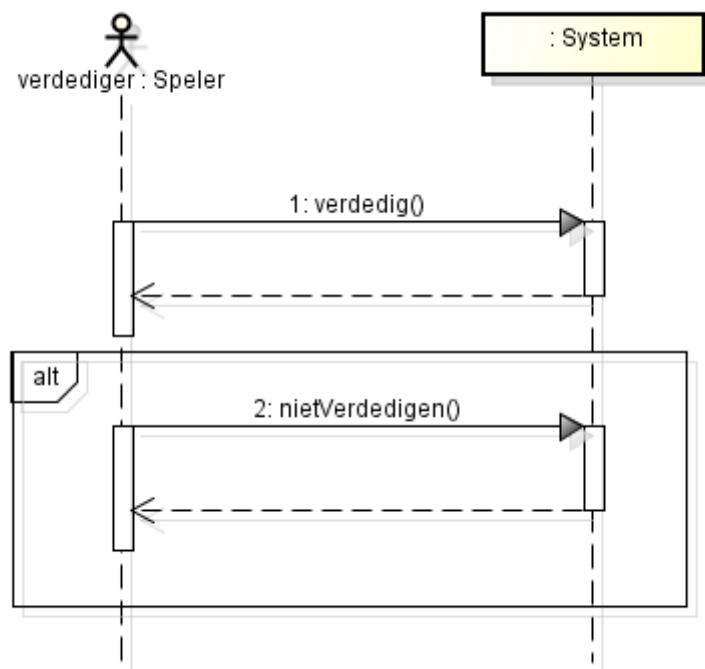
### 8.3 Kussen



### 8.4 Aanvallen

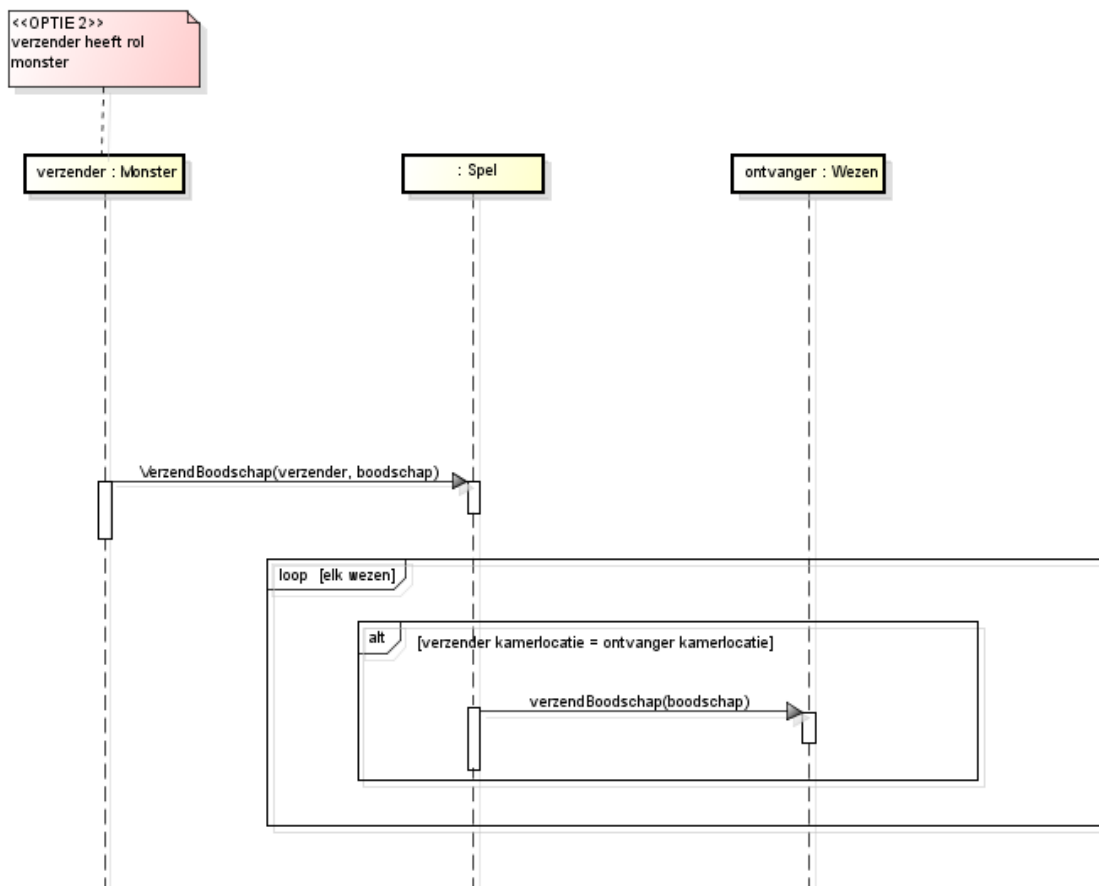
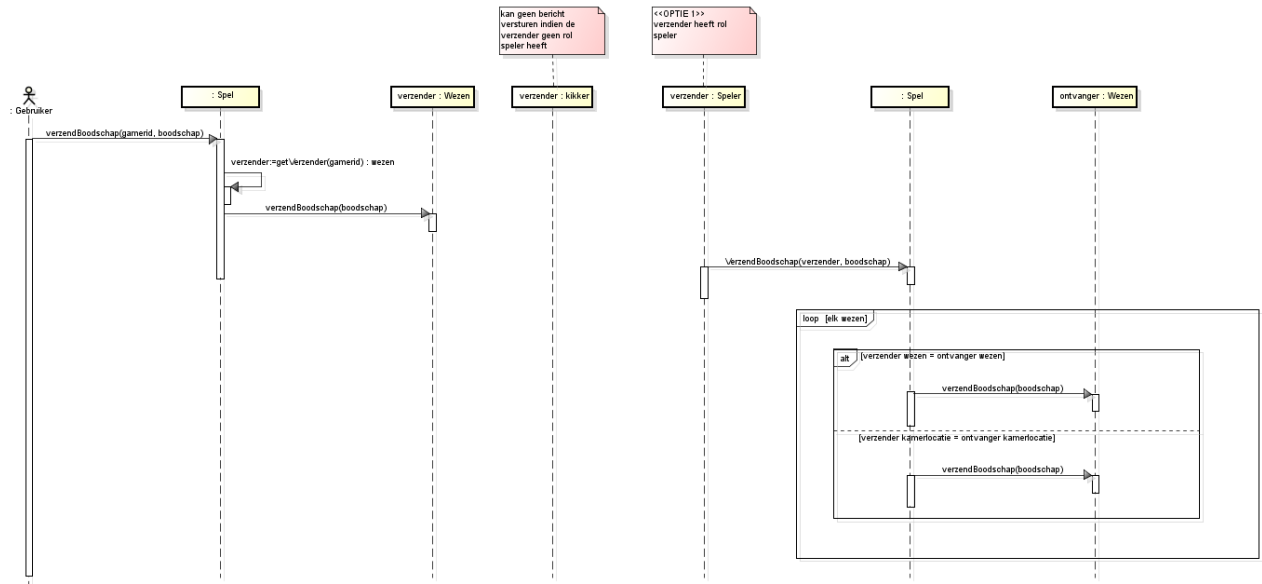


## 8.5 Verdedigen

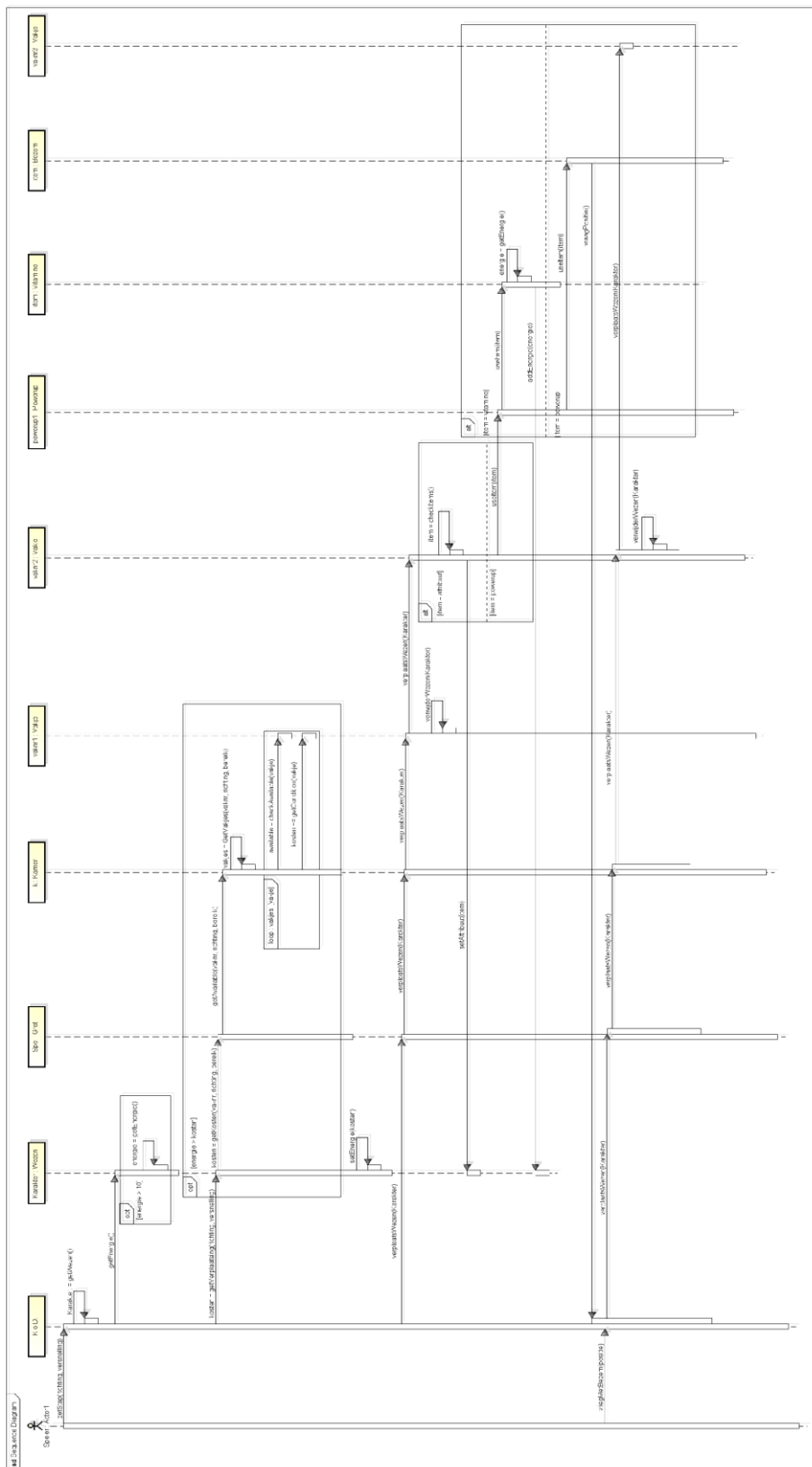


## 9 Sequence Diagrammen

### 9.1 Boodschap sturen

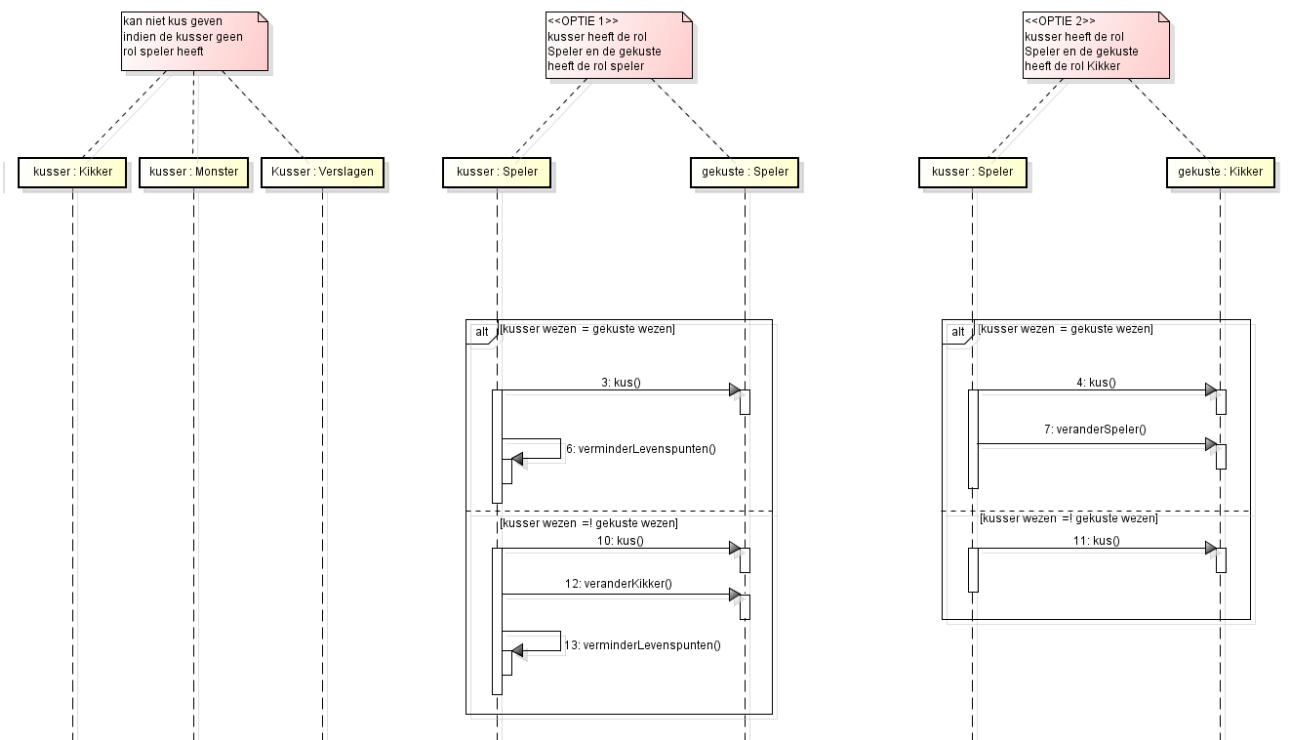
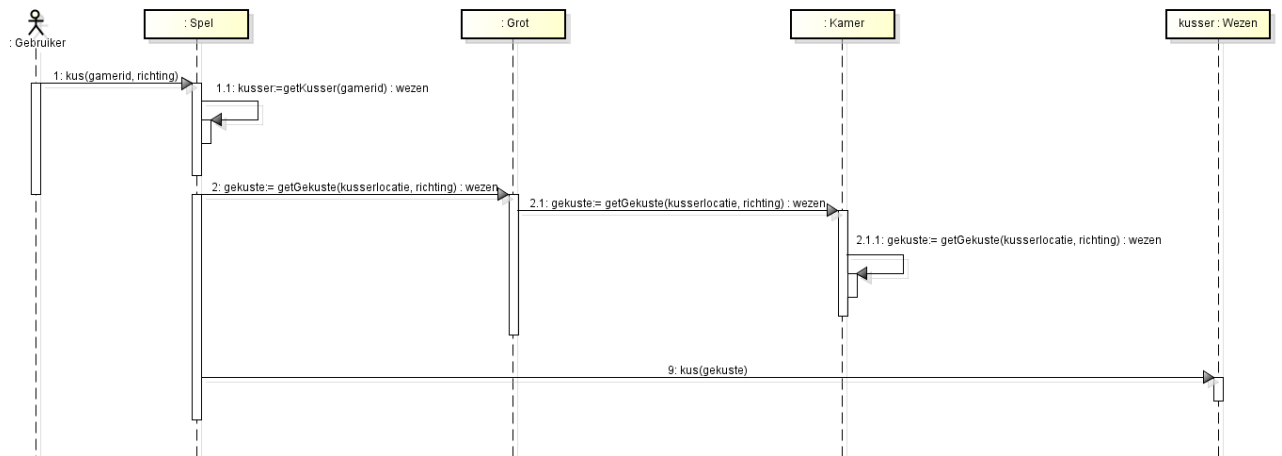


## 9.2 Bewegen

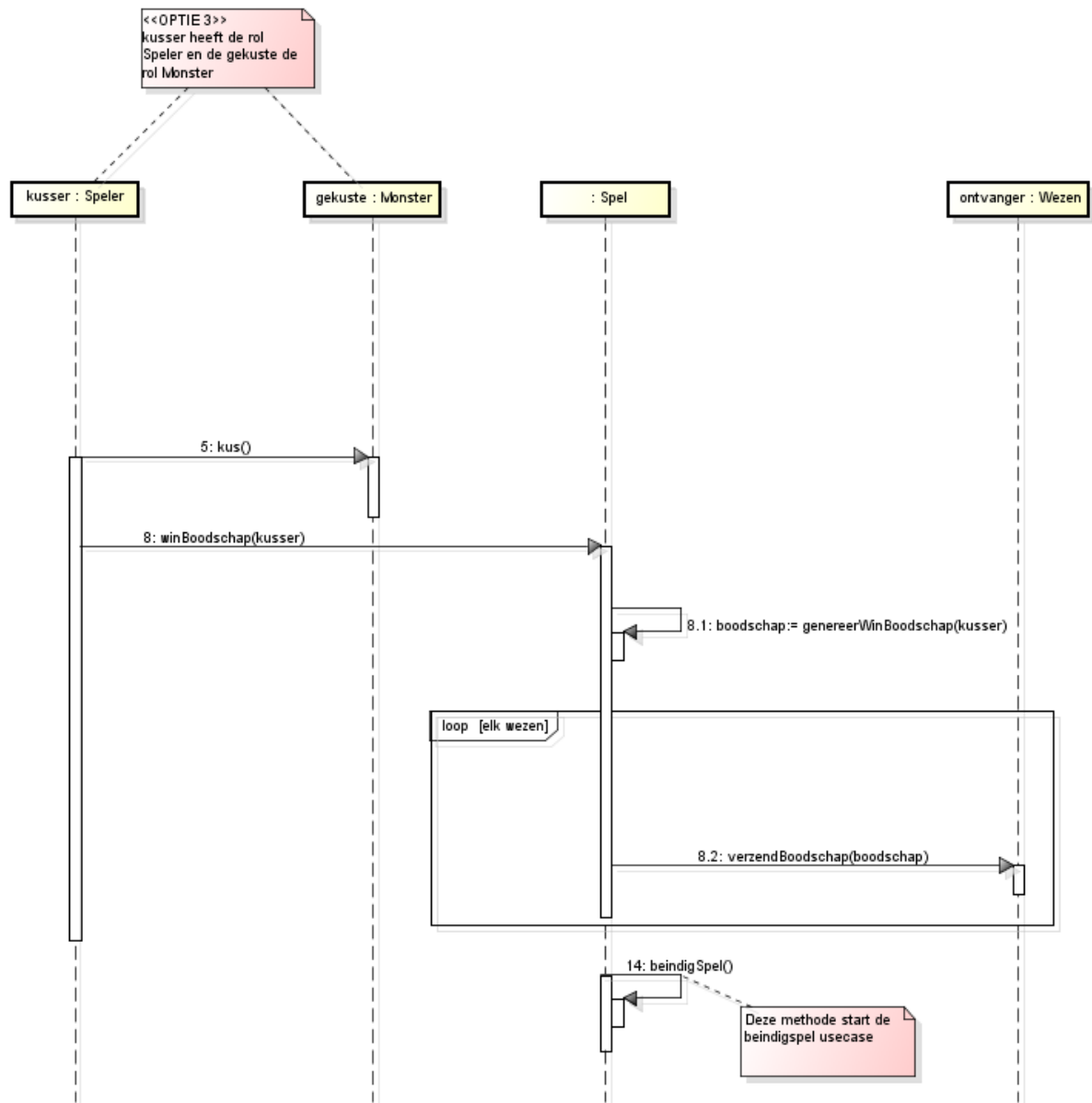


Bij het document is een origineel astah bestand en een png afbeelding meegeleverd. Het was geen optie om dit sequence diagram in delen op te splitsen.

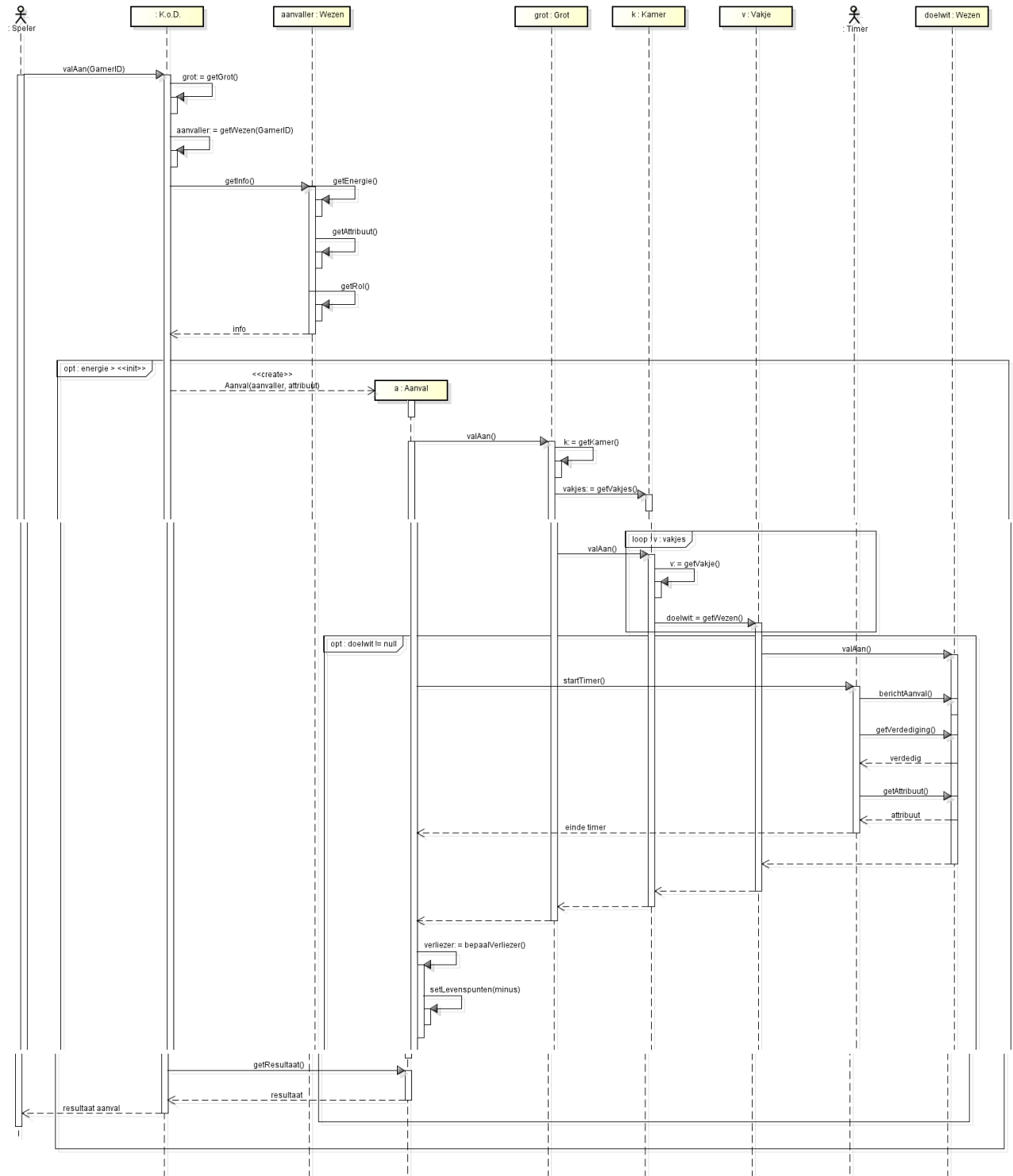
## 9.3 Kussen





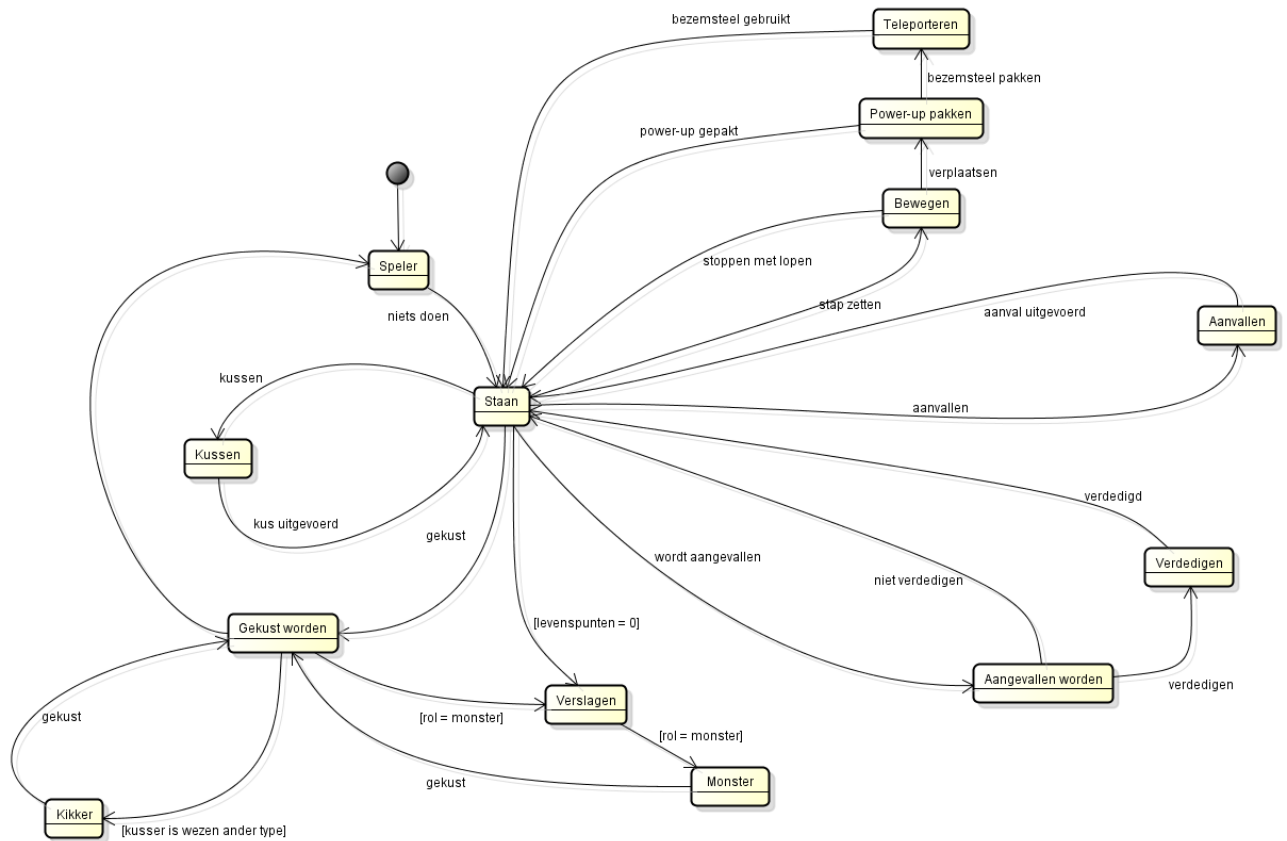


## 9.4 Aanvallen

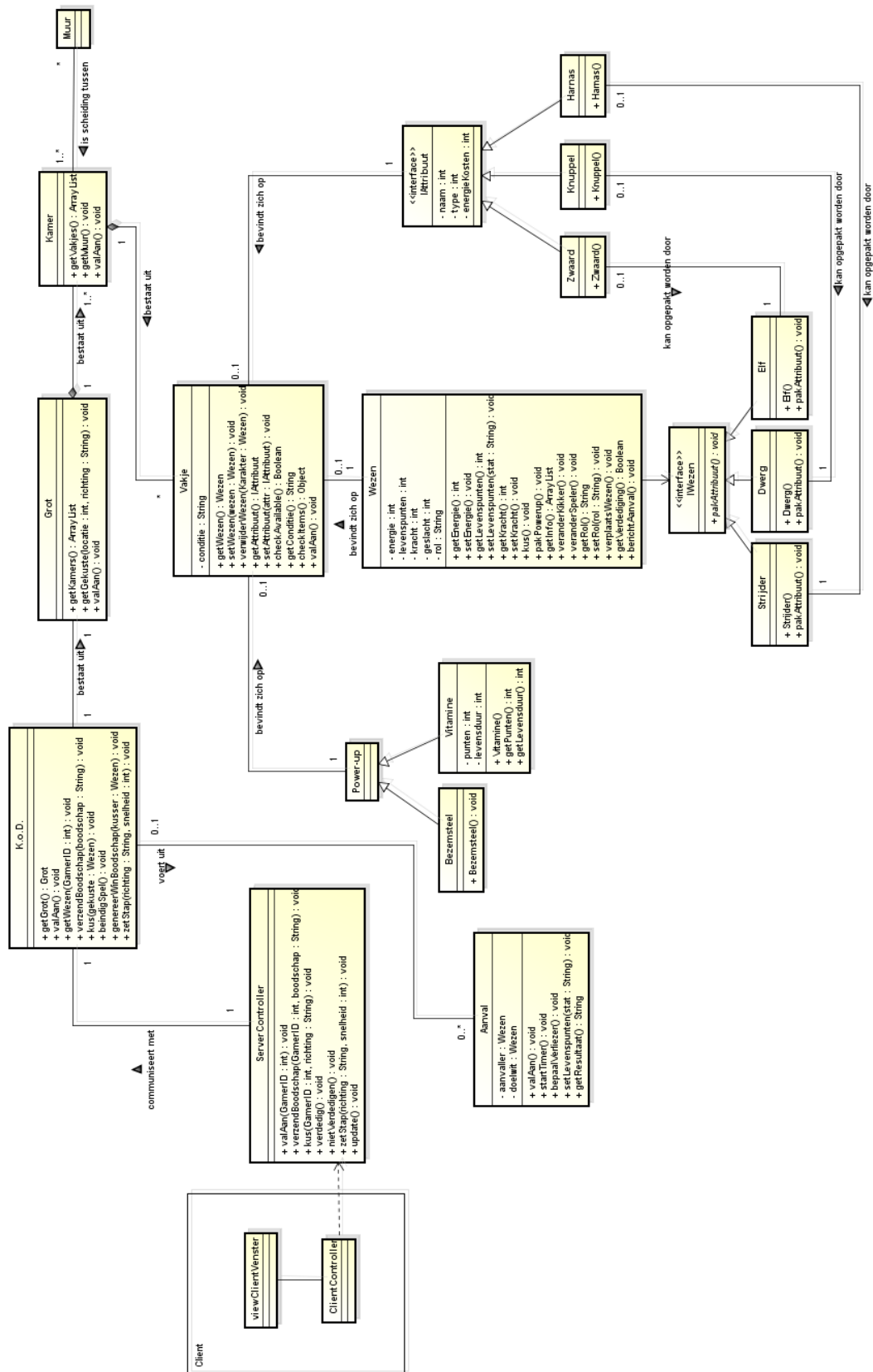


## 10 State Machine Diagram

In dit diagram zijn een aantal statussen van een wezen uitgewerkt. Een wezen kan wachten, kussen, gekust worden, verslagen worden, aangevallen worden en verdedigen, aanvallen, bewegen, power-ups pakken en teleporteren. Daarbij kan een speler ook nog veranderen in een kikker of een monster.



# 11 Design Class Diagram



## 12 Architectuurmodel

