

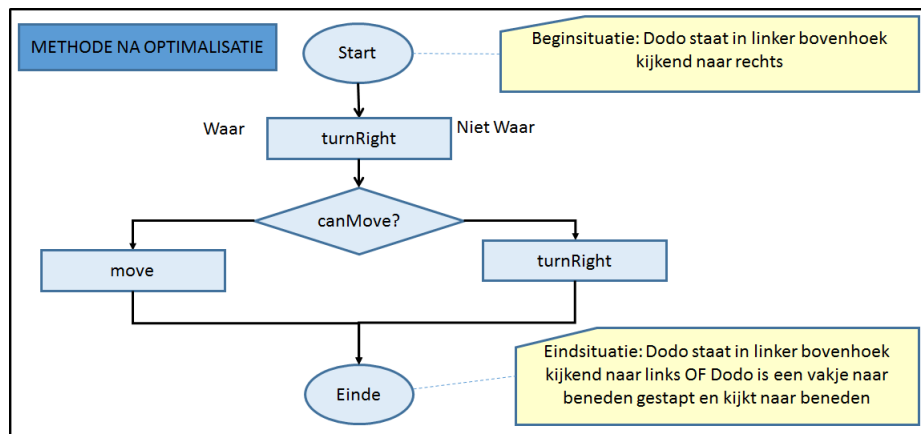
Opdracht 3: Uitwerkingen

– Objectgeoriënteerd Programmeren in Greenfoot –
Renske Smetsers-Weeda & Sjaak Smetsers

april, 2015

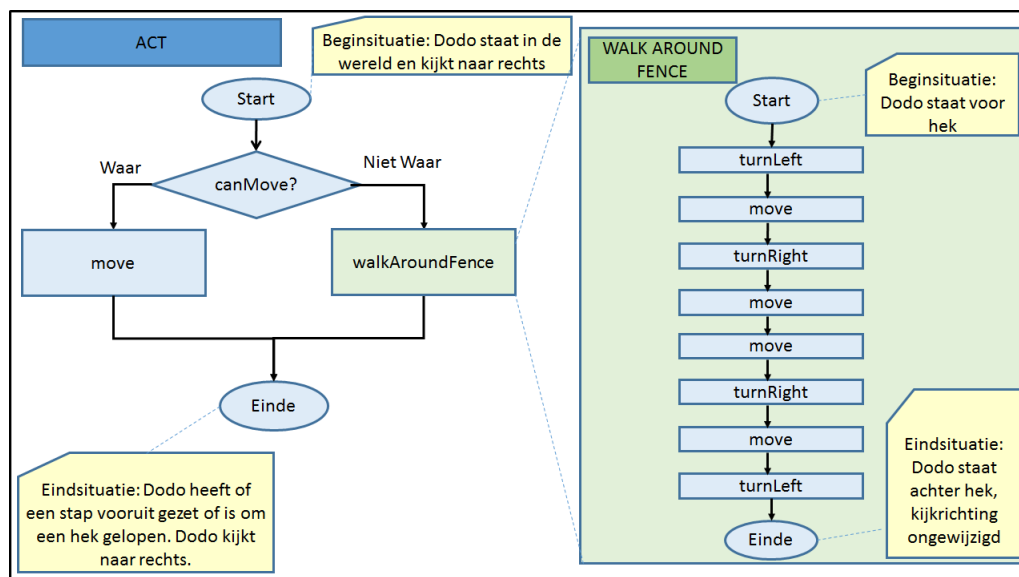
1 Voorbeelduitwerkingen van de opgaven

- Opgave 4.1.1 Optimaliseren doubleurs



Figuur 1: Stroomdiagram na optimalisatie verwijderen doubleurs

- Opgave 4.2.1 Submethodes gebruiken



Figuur 2: Stroomdiagram na optimalisatie met submethode

/**

* If possible, take a step forward

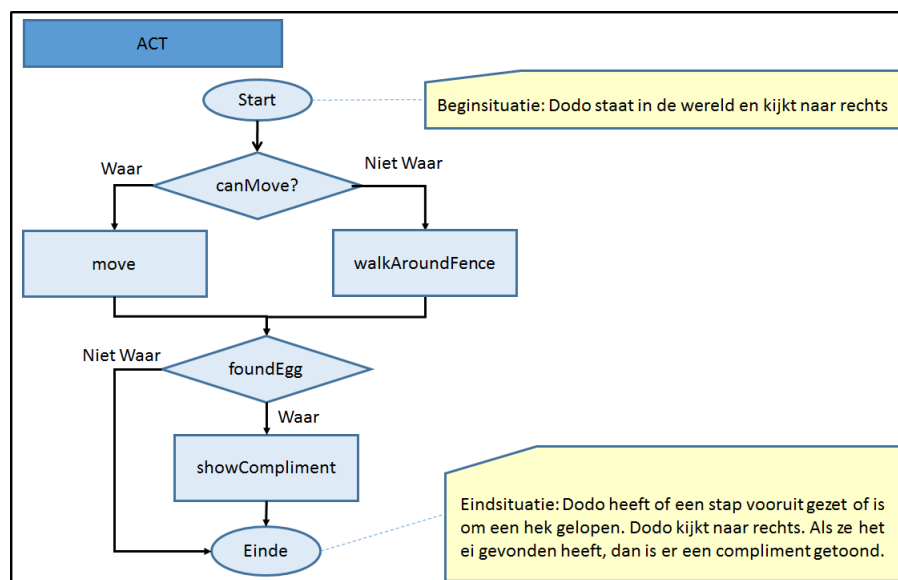
```

    * Otherwise, walk around the fence
    */
    public void act() {
        if ( canMove() ){
            move();
        }else{
            walkAroundFence();
        }
    }

    /**
     * Walk around the fence
     */
    public void walkAroundFence() {
        turnLeft();
        move();
        turnRight();
        move();
        move();
        turnRight();
        move();
        turnLeft();
    }
}

```

- Ei vinden met meerdere obstructies
Door 2 keer op 'Act' te drukken zou Mimi gewoon om beide hekjes moeten lopen. Als je op 'Run' klikt blijft ze doorlopen. We hebben namelijk geen conditie om te stoppen als ze een ei gevonden heeft.
- Complimentje geven



Figuur 3: Stroomdiagram Act met complimentje na vinden ei

```

/**
 * If possible, take a step forward

```

```
* Otherwise, walk around the fence
* If egg is found, show a dialog with a compliment
*/
public void act() {
    if ( canMove() ){
        move();
    }else{
        walkAroundFence();
    }

    if( foundEgg() ){
        showCompliment("Good job, egg found");
    }
}
```

- Programma stoppen

```
OPGAVE UITWERKING STOP PROGRAMMA
/**
* If possible, take a step forward
* Otherwise, walk around the fence
* If egg is found, show a dialog with a compliment and stop the program
*/
public void act() {
    if ( canMove() ) {
        move();
    } else {
        walkAroundFence();
    }

    if( foundEgg() ) {
        showCompliment("Good job, egg found");
        Greenfoot.stop();
    }
}
```