

Opdracht 4: Uitwerkingen

– Objectgeoriënteerd Programmeren in Greenfoot –

Renske Smetsers-Weeda & Sjaak Smetsers

mei, 2015

1 Voorbeelduitwerkingen van de opgaven

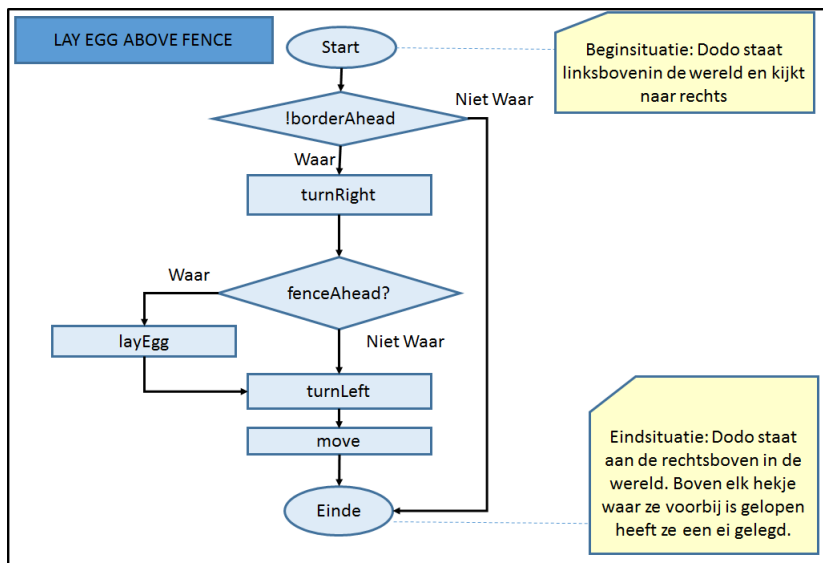
- Eigen functionaliteit: aparte methode

```
/**
 * find the egg
 */
public void act() {
    findEggBehindFence();
}

/*
 * If possible, take a step forward
 * If there is a fence in the way, walk around it
 * If egg is found, show a dialog with a compliment
 */
public void findEggBehindFence() {
    if ( canMove() ){
        move();
    } else {
        walkAroundFence();
    }

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

- Eitje boven hek leggen



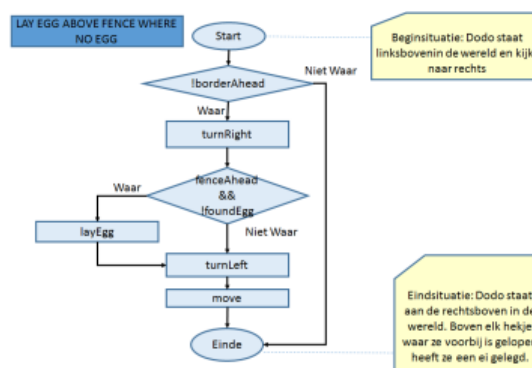
Figuur 1: Antwoordmodel voor Eitje boven hek leggen

```

/**
 * Walk from the top lefthand corner to the top
 * right hand corner of the world
 * If you find yourself standing above a piece
 * of fence, lay an egg there.
 */
public void layEggAboveFence() {
    if (! borderAhead() ) {
        turnRight();
        if ( fenceAhead() ) {
            layEgg();
        }
        turnLeft();
        move();
    }
}

```

- Geen dubbele eitjes



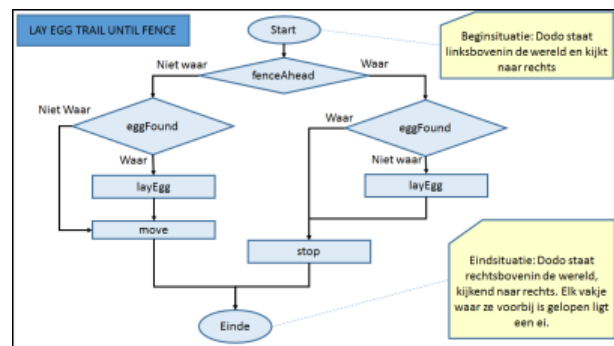
Figuur 2: Antwoordmodel voor Eitje boven hek leggen waar nog geen eitje ligt

```

/**
 * Walk from the top lefthand corner to the
 * top right hand corner of the world
 * If you find yourself standing above a
 * piece of fence and there is no egg there, lay an egg there.
 */
public void layEggAboveFenceWhereNoEgg( ) {
    if ( ! borderAhead( ) ) {
        turnRight( );
        if ( fenceAhead( ) && !foundEgg( ) ){
            layEgg( );
        }
        turnLeft( );
        move( );
    }
}

```

- Spoor van eitjes



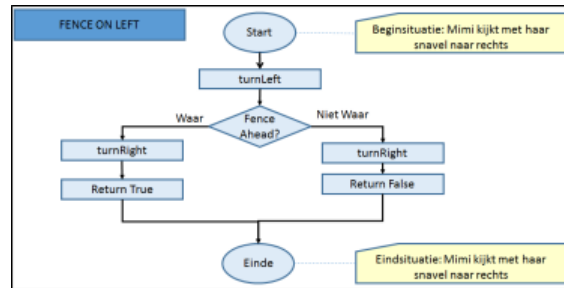
Figuur 3: Antwoordmodel voor leg spoor van eitjes tot hek

```

/**
 * Walk from the top lefthand corner towards
 * the right hand corner until a fence is reached
 * Leave a trail of eggs behind as you go. If
 * there is already an egg, don't lay another one.
 */
public void layEggTrailUntilFence( ) {
    if ( fenceAhead( ) ) {
        if( ! foundEgg( ) ) {
            layEgg();
        }
        showCompliment("I'm done");
        Greenfoot.stop();
    } else {
        if( foundEgg( ) ) {
            move();
        } else {
            layEgg();
            move();
        }
    }
}

```

- Loop door een tunnel



Figuur 4: Antwoordmodel voor fence on left

```

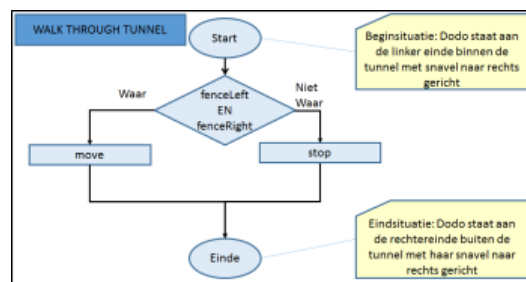
/*
 * Check if there is a fence on the left-hand side
 */

public boolean fenceOnLeft() {
    turnLeft();
    if(fenceAhead() ) {
        turnRight();
        return true;
    } else {
        turnRight();
        return false;
    }
}

/*
 * Check if there is a fence on the right-hand side
 */

public boolean fenceOnRight() {
    turnRight();
    if( fenceAhead() ) {
        turnLeft();
        return true;
    } else {
        turnLeft();
        return false;
    }
}

```



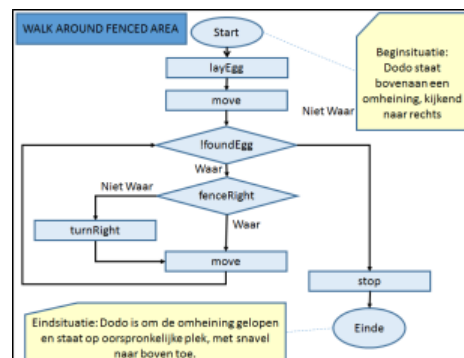
Figuur 5: Antwoordmodel voor walk through tunnel

```

iiiiii HEAD
=====
/**
 * Walk through tunnel and stop at end
 */
public void walkThroughTunnel() {
    if ( fenceOnLeft() && fenceOnRight() ) {
        move();
    } else {
        Greenfoot.stop();
    }
}
}

```

- Om een omheining lopen



Figuur 6: Antwoordmodel voor walk around fenced area

```

iiiiii HEAD =====
iiiiii f6245ae6672f82e747c9093af0e450964c4b78de

/**
 * Starting at the top of a fenced area,
 * Lay an egg
 * Walk around a fenced area, with the fence on your
 * right-hand side until you reach the egg again
 * Then stop.
 */
public void walkAroundFencedArea( ) {
    layEgg(); // leg een ei om begin positie vast te leggen
    move();

    while( !foundEgg() ){
        // if(!foundEgg() ){
        if( !fenceOnRight() ){
            turnRight();
            move();
        } else {
            if( !fenceAhead() ){
                move();
            } else {
                if ( !fenceOnLeft() ){
                    turnLeft();
                    move();
                }
            }
        }
    }
}

```

```

        }
        else{
            // dead-end
        }
    }
}
}
hatchEgg(); // ruim eitje weer op
//showCompliment("I walked around the fence!");
Greenfoot.stop();
}

```

- Eitjes volgen tot de nest
Boolean methodes:

```

/*
 * Check if there is a egg on the left-hand side
 */

public boolean eggOnLeft(){
    turnLeft();
    if(eggAhead() ){
        turnRight();
        return true;
    }else{
        turnRight();
        return false;
    }
}

/*
 * Check if there is a egg on the right-hand side
 */

public boolean eggOnRight(){
    turnRight();
    if(eggAhead() ){
        turnLeft();
        return true;
    }else{
        turnLeft();
        return false;
    }
}

/*
 * Check if there is a egg on behind
 */

public boolean eggBehind(){
    turnRight();
    turnRight();
    if(eggAhead() ){
        turnLeft();
        turnLeft();
        return true;
    }else{
        turnLeft();

```

```

        turnLeft();
        return false;
    }
}

/*
 * Check if the nest is in the cell ahead.
 */

public boolean nestFound(){
    if( nestAhead() ){
        return true;
    }else{
        turnRight();
        if( nestAhead() ){
            return true;
        }else{
            turnRight();
            if( nestAhead() ){
                return true;
            }else{
                turnRight();
                if( nestAhead() ){
                    return true;
                }
            }
        }
    }
    turnRight();
    return false;
}

```

- Methode om spoor van eieren te volgen voor het vinden van de nest (werkt in beide werelden)

```

/*
 * Follow a trail of eggs until.
 * When the nest is reached, show a compliment dialog and stop.
 */
public void followEggTrailUntilNest(){
    if(nestFound() ){
        showCompliment("nest found!");
        Greenfoot.stop();
    }else{
        if ( eggAhead() ) {
            move();
        }else{
            if ( eggOnRight() ){
                turnRight();
                move();
            }else{
                if ( eggOnLeft() ){
                    turnLeft();
                    move();
                }else{
                    if( eggBehind() ){
                        turnLeft();
                        turnLeft();
                    }
                }
            }
        }
    }
}

```

```
        move();
    }else{
        showError("I can't find the nest!");
        Greenfoot.stop();
    }
}
}
```

- Dolhof

```
public void walkThroughMaze(){
    if( nestFound() ) {
        showCompliment("nest found!");
        Greenfoot.stop();
    } else {
        if (!fenceOnRight() ){
            turnRight();
            move();
        }else{
            if( !fenceAhead() ){
                move();
            }else{
                if( !fenceOnLeft() ){
                    turnLeft();
                    move();
                }else{ //turn around
                    turnLeft();
                    turnLeft();
                    move();
                }
            }
        }
    }
}
```

Mogelijke alternatieve strategieën:

- laat een spoor van eitjes achter.
- draai om als je een doodlopend pad bent ingeslagen.
- als je kunt kiezen tussen een ei-vrij pad of een pad met eieren, kies het pad met eieren