

Rapport

HAN – UNIVERSITY OF APPLIED SCIENCES

Johnny Chen (668003) | HBO-ICT | Studiejaar 1 Docentbegeleider: Martijn Driessen

Klas: ITA-1C

Course: SPAD

Datum: 22-10-2021

Versie: Castle Bomber

Inhoudsopgave

[Oplossing algoritmen 3](#_Toc85809637)

[Maak spelbord 3](#_Toc85809638)

[Array shuffle 4](#_Toc85809639)

[Gooi bom 4](#_Toc85809640)

[Software architectuur 5](#_Toc85809641)

[Reflectie op codekwaliteit 7](#_Toc85809642)

[Testrapportage 8](#_Toc85809643)

[Koning 8](#_Toc85809644)

[Een 1-player game 8](#_Toc85809645)

[Bom 8](#_Toc85809646)

[1 type schat 8](#_Toc85809647)

[Schatten en bommendepots willekeurig 8](#_Toc85809648)

[Elementen 9](#_Toc85809649)

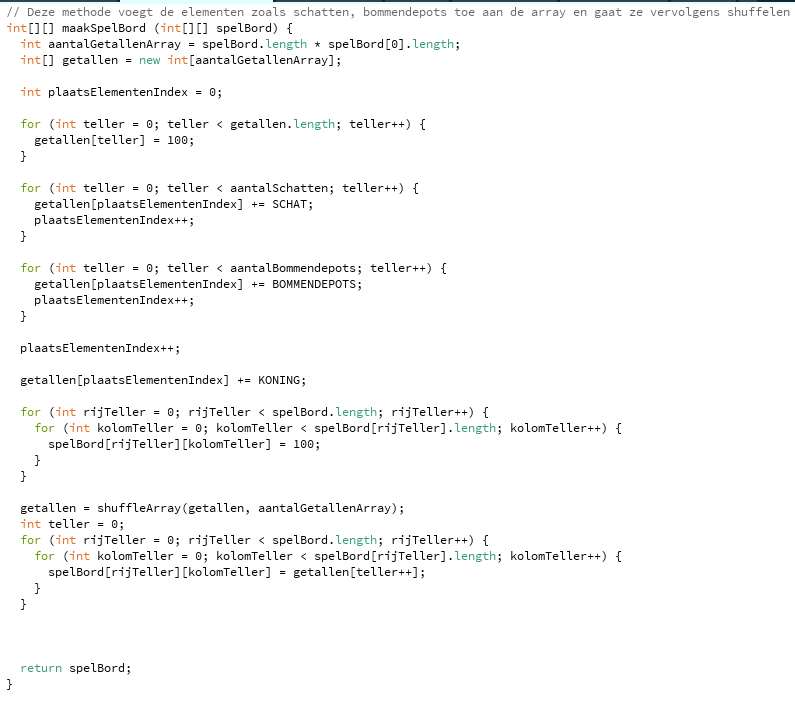
[Score 9](#_Toc85809650)

[Blootleggen bommendepot 9](#_Toc85809651)

[Instelbare speelveld 9](#_Toc85809652)

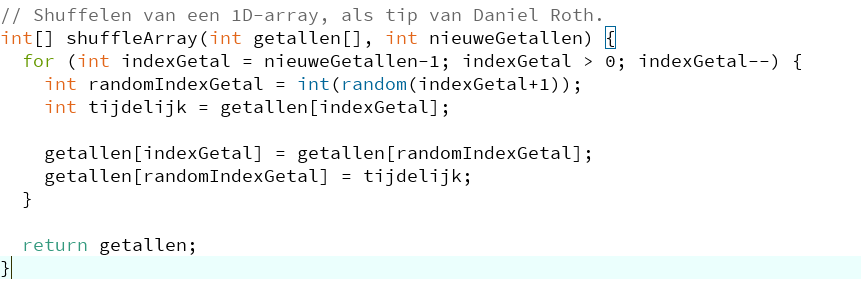
# Oplossing algoritmen

## Maak spelbord



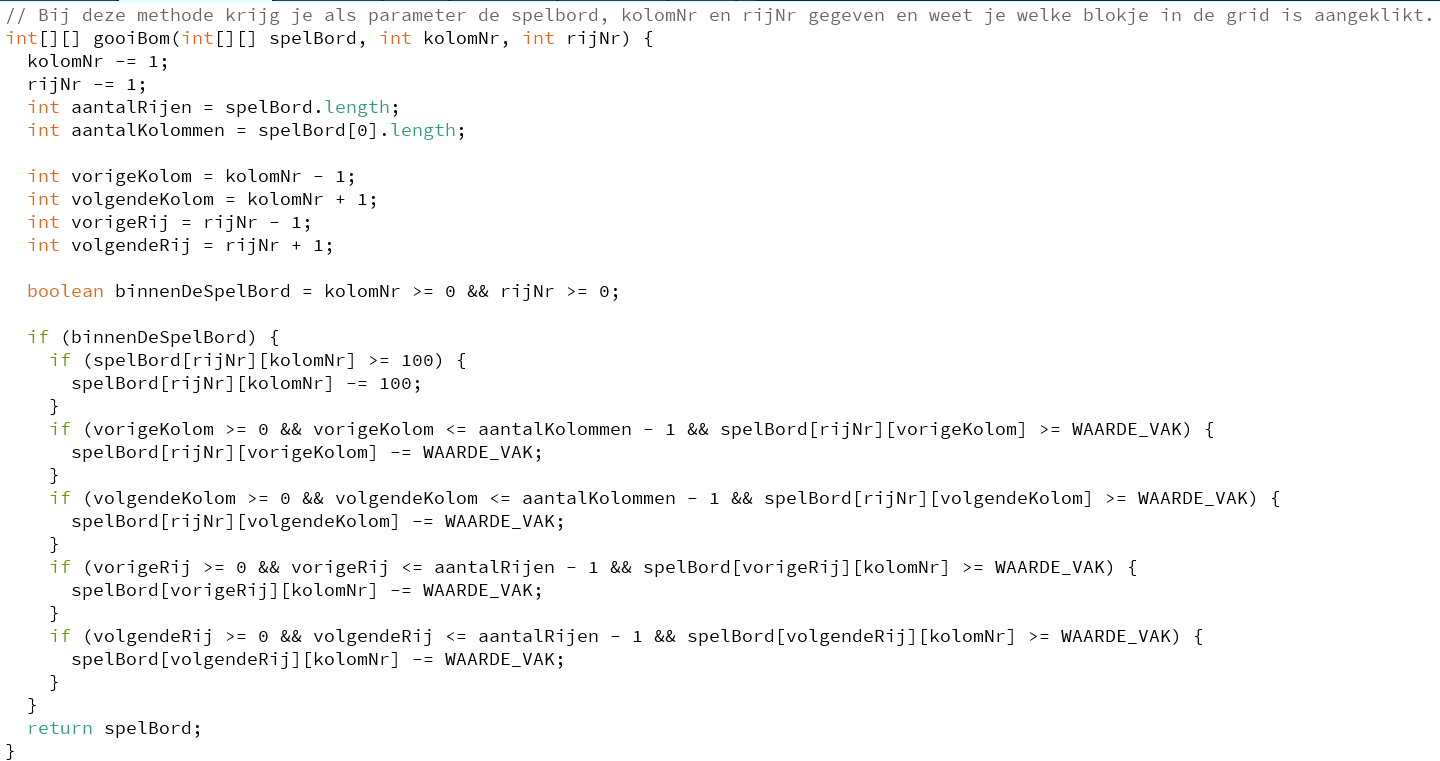
Dit is een van de drie belangrijke oplossingalgoritmen. Bij deze methode maak ik een spelbord aan, waarbij een spelbord is aangegeven via de parameter. Vervolgens maak ik een 1D array aan van getallen met de totale grootte van de spelbord. Bij de eerste for-loop voeg ik bij alle getallen 100 toe, daarna voeg ik bij de volgende for-loop alle schatten toe en dan nog de bommendepots. Als laatst voeg ik de koning toe en heb ik nu al mijn elementen verwerkt in een 1D array. Deze 1D array shuffle ik vervolgens, waardoor ik deze 1D array kan gebruiken voor mijn spelbord. Ik verwerk de nieuwe 2D array in mijn spelbord, waardoor ik nu alleen maar random elementen krijgen per verschillende blok.

## Array shuffle



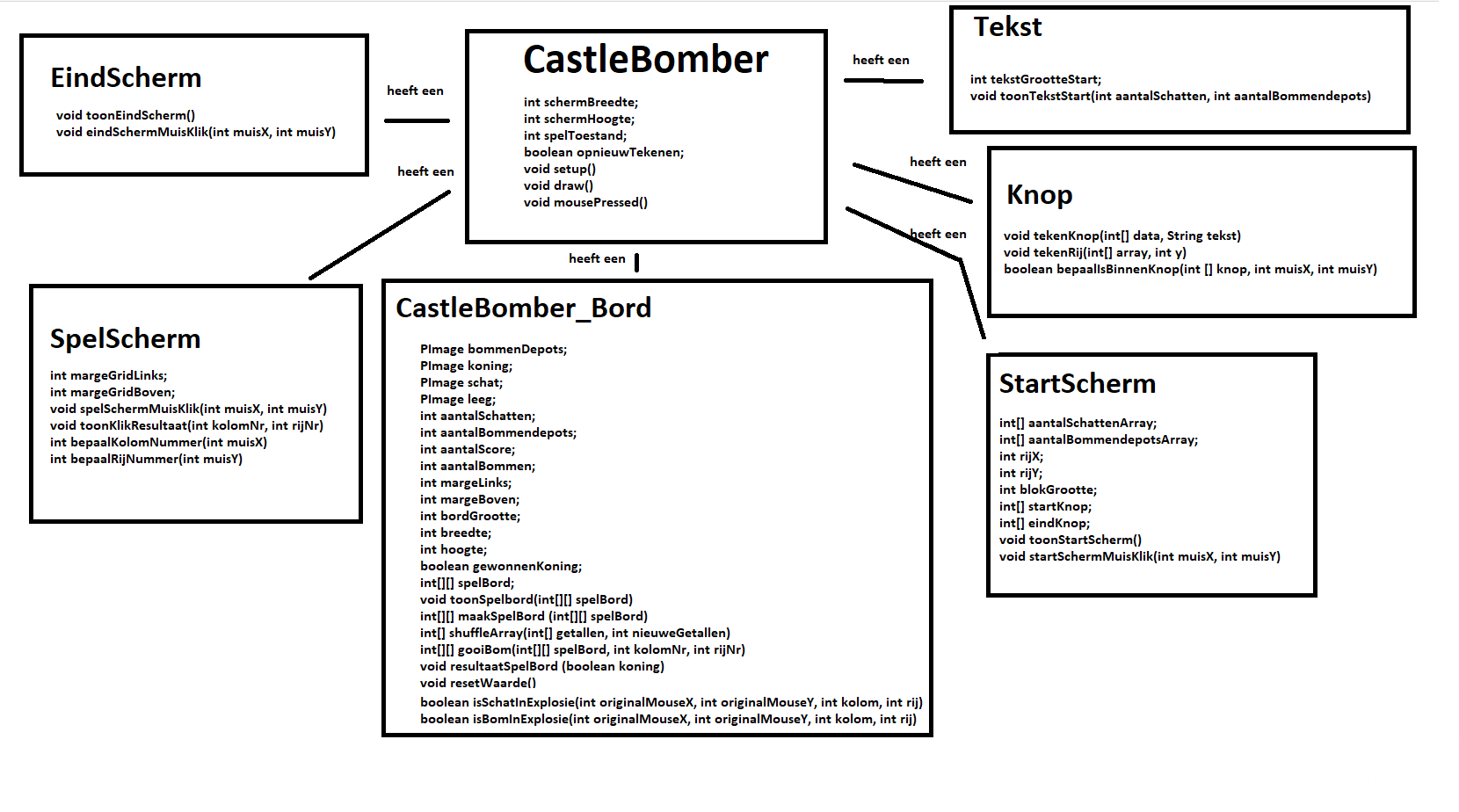
Bij deze methode shuffle ik de array die is aangegeven bij de methode maakSpelBord. Je geeft hierbij de 1D array en de totale grootte van de 2D aan. Vervolgens husselt de for-lus alle getallen in de array, met als gevolg dat de array een compleet andere volgorde is. Zonder de array shuffle heeft het spelbord altijd hetzelfde volgorde en dat willen we niet hebben. Deze array shuffle heb ik van Daniel Roth als tip meegekregen en het werkt gelukkig goed.

## Gooi bom



Na het tekenen van het spelbord had ik nog als probleem dat ik geen bommen kan plaatsen. Hierbij had ik vervolgens de gooiBom methode gemaakt. Je krijgt als eerst de spelbord array, kolomnummer en rijnummer aangegeven. De kolomnummer en rijnummer krijg je aangegeven door de methode bepaalKolomNummer en bepaalRijNummer. Als eerst doe ik bij zowel kolomNr als bij rijNr een één eraf. De reden hiervoor is omdat de index bij 0 beginnen. Als ik dit niet deed dan plaats ik de schatten ergens anders dan waar ik op had geklikt. Vervolgens controleer ik of de klik wel binnen het spelbord ligt en heb ik 5 if-statements gemaakt. Met deze if-statements haal ik WAARDE\_VAK eraf als het deze conditie voldoet. Met die waardes doe ik met een andere methode een afbeelding vormen, met als gevolg dat het een plusje vormt als ik op een vakje klik in het spelbord.

# Software architectuur



|  |  |  |
| --- | --- | --- |
| Module | Events, methodes en variables | Toelichting |
| Events | void setup() | Background wordt gewist en afbeeldingen worden hier geladen. |
|  | void draw() | Het scherm wordt hier getekend als opnieuwTekenen = true; afhankelijk van de speltoestand. |
|  | void mousePressed() | Alle muisklikken worden hier opgevangen, bijvoorbeeld wanneer je op een knop klikt of op het spelbord. |
| CastleBomber\_Bord | void toonSpelbord(int[][] spelBord) | Deze methode vertoont hoe de spelbord er moet gaan uitzien |
|  | boolean isSchatInExplosie(int originalMouseX, int originalMouseY, int kolom, int rij) | Bekijk of schat in gooiBom zit |
|  | boolean isBomInExplosie(int originalMouseX, int originalMouseY, int kolom, int rij) | Bekijk of bom in gooiBom zit |
|  | int[][] maakSpelBord (int[][] spelBord) | Deze methode voegt de elementen zoals schatten, bommendepots toe aan de array en gaat ze vervolgens shuffelen |
|  | int[] shuffleArray(int[] getallen, int nieuweGetallen) | Het shuffelen van de array voor het spelbord. |
|  | int[][] gooiBom(int[][] spelBord, int kolomNr, int rijNr) | Bij deze methode krijg je als parameter de spelbord, kolomNr en rijNr gegeven en weet je welke blokje in de grid is aangeklikt |
|  | void resultaatSpelBord (boolean koning) | Tekst voor eindscherm |
|  | void resetWaarde() | Reset alle waarde, zodat hij met default instellingen weer door kan gaan en nieuwe met de getallen van vorige game |
| Eindscherm | void toonEindScherm() | Het eindscherm wordt vertoond door de switch case |
|  | void eindSchermMuisKlik(int muisX, int muisY) | Verwerkt een muisklik en bepaalt vervolgens of het de eindknop heeft geraakt. |
| Knop | void tekenKnop(int[] data, String tekst) | Deze methode ontvangt een 1D array en een string als parameter en tekent vervolgens een knop |
|  | void tekenRij(int[] array, int y) | Deze methode ontvangt een array en een int en tekent vervolgens een rij van knoppen |
|  | boolean bepaalIsBinnenKnop(int [] knop, int muisX, int muisY) | Bij deze methode krijg je een return van true of false als de x en y coördinaten binnen een bepaalde knop zitten |
| Spelscherm | void toonSpelScherm() | Het spelscherm tonen door de switch case |
|  | void spelSchermMuisKlik(int muisX, int muisY) | Verwerkt de muisklik in het spelscherm |
|  | void toonKlikResultaat(int kolomNr, int rijNr) | Deze methode laat zien welke rijnummer en kolomnummer het zijn. Bij ongeldige klik, komt er een ongeldige rij of kolom tevoorschijn |
|  | int bepaalKolomNummer(int muisX) | Deze methode return een kolomnummer |
|  | int bepaalKolomNummer(int muisX) | Deze methode return een rijnummer |
| Startscherm | void toonStartScherm() | De startscherm tonen door de switch case |
|  | void startSchermMuisKlik(int muisX, int muisY) | Verwerkt een muisklik in het startscherm en kijkt vervolgens of je hebt geklikt op de startknop |
| Tekst | void toonTekstStart(int aantalSchatten, int aantalBommendepots) | Vertoont de tekst die nodig is in het startscherm |

# Reflectie op codekwaliteit

Voor mijn gevoel vind ik dat mijn codekwaliteit voldoet aan kwaliteitseisen die we hebben gerealiseerd bij SBP en SPAD. Als eerst heb ik mijn methode zo verdeeld dat hij maar één verantwoordelijkheid heeft per methode.

Hierbij heb ik goed gebruik gemaakt van parameters die belangrijk zijn voor de methode, zodat de methode goed werkt. Bij een paar methodes heb ik natuurlijk ook een return waarde. Het is belangrijk om een paar return waarde hebben en niet allemaal een void. Ik heb als tip gekregen dat void meer voor het tekenen is en geen andere dingen.

Al mijn methode heb ik ook in een lowerCamelCase getekend en bevat er natuurlijk een werkwoord. Daarnaast heb ik mij best proberen te doe al die methode toe te passen aan een bepaalde module. Soms was het moeilijk, doordat de variable in een andere module eigenlijk niet gebruikt kan worden dus het was een beetje puzzelen. Alle modules heb ik in UpperCamelCase geschreven, maar het moeilijkste gedeelte was toch wel dat je soms variable van een andere module nodig hebt.

# Testrapportage

## Koning

Als ik een keer gewonnen heb door middel van de koning te kiezen. De game daarna heb ik verloren, wat voor tekst komt er dan?

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Win door de koning | Je hebt gewonnen, koning is gevonden | Ja |
| 2 | Verlies door bommen | Je hebt verloren | Nee |

Oplossing : gewonnenKoning = false; in resultaatSpelBord(boolean koning)

## Een 1-player game

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Klik op startknop | Een 1-player game | Ja |
| 2 | Klik buiten startknop | Er gebeurt niks | Ja |

## Bom

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Klik op spelbord | Bom is geplaats door middel van een plus | Ja |
| 2 | Klik buiten spelbord | Bom is niet geplaats, er gebeurt niks | Ja |

## 1 type schat

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Klik op spelbord | 1 type schat | Ja |

## Schatten en bommendepots willekeurig

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Klik op spelbord | Willekeurige schatten en bommendepots verdeeld over het spelbord | Ja |
| 2 | Klik op opnieuw | Het spelbord is weer verdeeld met willekeurige schatten en bommendepots | Ja |

## Elementen

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Klik op spelbord 2x | Er kunnen geen elementen op dezelfde plek staan | Ja |
| 2 | Klik op spelbord meer dan 2x | Geen elementen op dezelfde plek | Ja |

## Score

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Klikken op spelbord | Bommen en schat wordt bijgehouden | Ja |
| 2 | Klik buiten spelbord | Bommen en schat wordt bijgehouden | Ja |

\*\* Bom kan wellicht buiten spelbord worden geplaats door een misclick. Eigen schuld dus

## Blootleggen bommendepot

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Klik op spelbord | Bommen verhoogt | Ja |
| 2 | Klik 2x of meer op spelbord | Bommen blijven hetzelfde | Nee |

## Instelbare speelveld

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase** | **Actie** | **Verwacht resultaat** | **Geslaagd** |
| 1 | Aanpassen van de variable van het speelveld | Speelveld wordt verkleind/vergroot | Ja |