

Hogeschool Arnhem Nijmegen

# ONTWERP VAN COMMAND 2 CONQUER

619105 - Bas Slijkhuis

31-10-2018 V1.0 Docent: Fritz van Deventer

## Inhoudsopgave

1. Inleiding .....	2
2. De globale flow .....	3
3. Schermontwerp .....	4
Startscherm .....	4
Gamemode scherm .....	4
Levelscherms .....	5
Eerste beurt .....	5
Hogere scores maken .....	6
Bom raken .....	6
Eindscherms .....	7
4. Softwarebeschrijving .....	8
Globale software structuur .....	8
Events, methode en variabelen .....	8
Samenhang van de events, methoden en variabelen met de ontwerpen .....	10
De programma flow .....	11
5. Testcases .....	13
Testcase 1: kiezen tussen spel stoppen of starten .....	13
Testcase 2: kiezen tussen gamemodes .....	13
Testcase 3: het spel spelen .....	14
Testcase 4: kiezen tussen spel stoppen of opnieuw spelen .....	14

## 1. Inleiding

Het spel dat ik heb gemaakt is een multiplayer spel, waar je het op neemt tegen de andere speler om te proberen om het spel te winnen met het meeste aantal punten. Dit doe je door zoveel mogelijk vakjes te krijgen en aanliggende vakjes geven extra punten. Maar je moet oppassen met bommen, deze zorgen dat je beurt over is en dat je een minpunt krijgt. Als alle vlakjes zijn geclaimd exclusief bommen is het spel af.

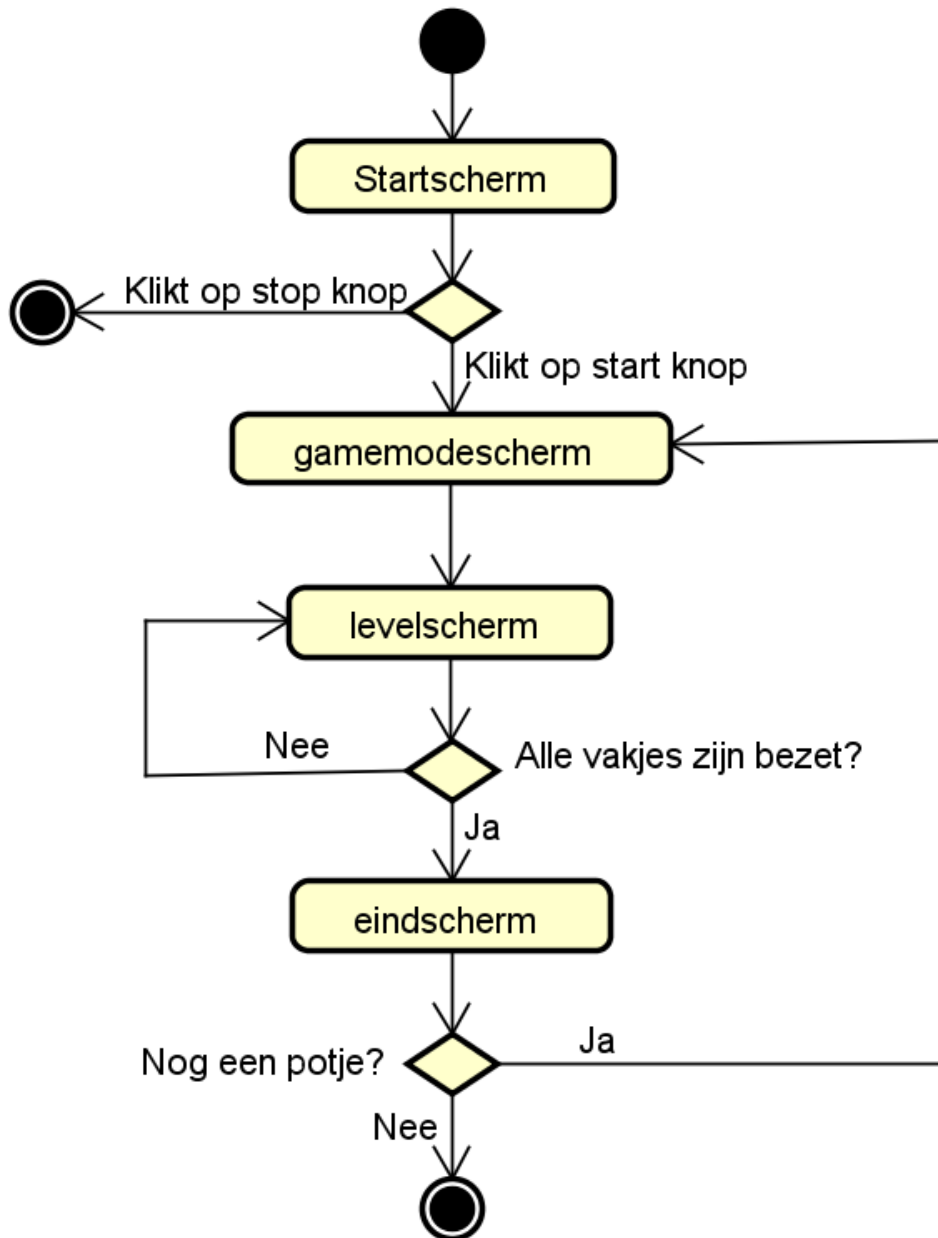
Je hebt de optie om tussen verschillende soorten groottes van het spel te kiezen. Je kunt kiezen uit 5 bij 5, 7 bij 7 en 10 bij 10. Bij elke gamemode is het aantal bommen 10% van het complete veld. Dus wanneer het een veld is van 5 bij 5 zijn het aantal bommen 5.

Aan de volgende eisen moet minimaal zijn voldaan:

- Het speelveld is instelbaar met velden van minimaal 5 bij 5 en maximaal 10 bij 10
- Het spel moet met 2 spelers kunnen worden gespeeld
- De kleuren van de spelers zijn willekeurig gekozen maar duidelijk verschillend
- De speler die mag beginnen is willekeurig gekozen, maar daarna is het om de beurt
- Een mijn zorgt voor een beurt wissel en dat het veld geen punten oplevert en het klikken op een mijn geeft de speler een minpunt
- De punten van de spelers staan constant op beeld
- Het dashboard geeft aan welke kleur welke speler is
- Resterende aantal mijnen wordt getoond
- Het spel reageert alleen op muisklikken op ongeopende velden
- Als het spel is afgelopen komt er een scherm met een optie om te stoppen of om nogmaals te spelen, waar ook komt te staan wie er heeft gewonnen en met hoeveel punten verschil.

## 2. De globale flow

Als we gaan kijken naar de globale flow zijn deze situaties mogelijk op globaal niveau. De pijlen geven aan in welke situatie de situatie kan overgaan. Alles in de globale flow wordt geactiveerd door dat een gebruiker iets doet. Bijvoorbeeld op een button klikken.

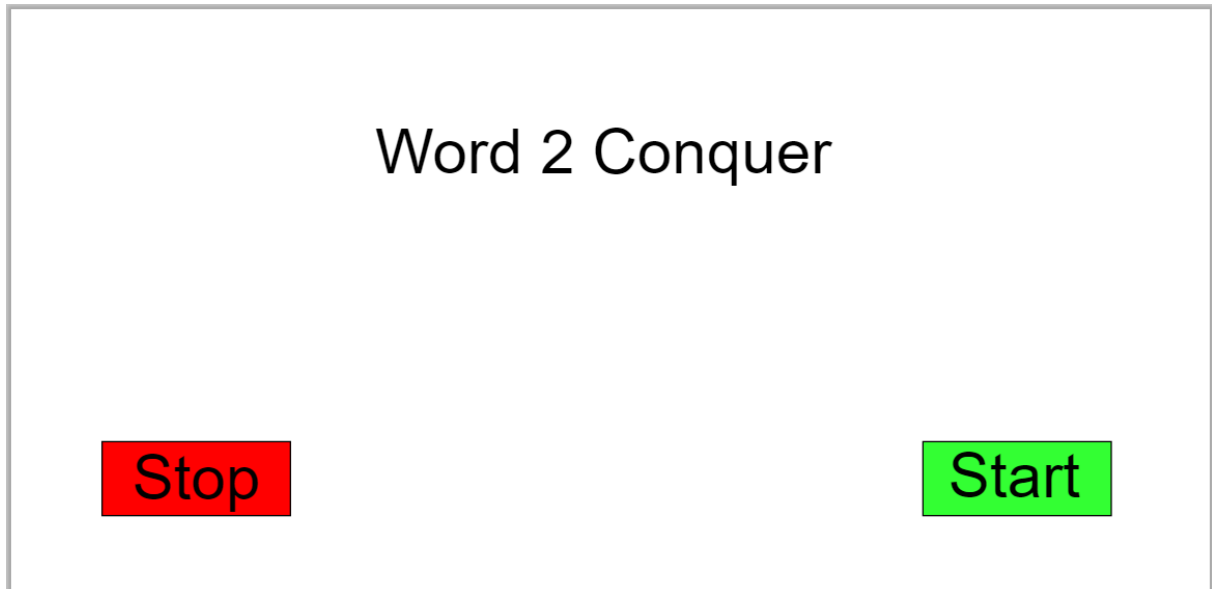


Het spel kan ter alle tijden worden beëindigd door de kruis knop, daarom wordt dit niet bij elk niveau weergegeven. Toch staat de eind knop twee keer wel weergegeven. Dit is omdat er dan in het programma zelf een stop knop zit. Hier weet het programma dus wel dat het programma moet stoppen.

### 3. Schermontwerp

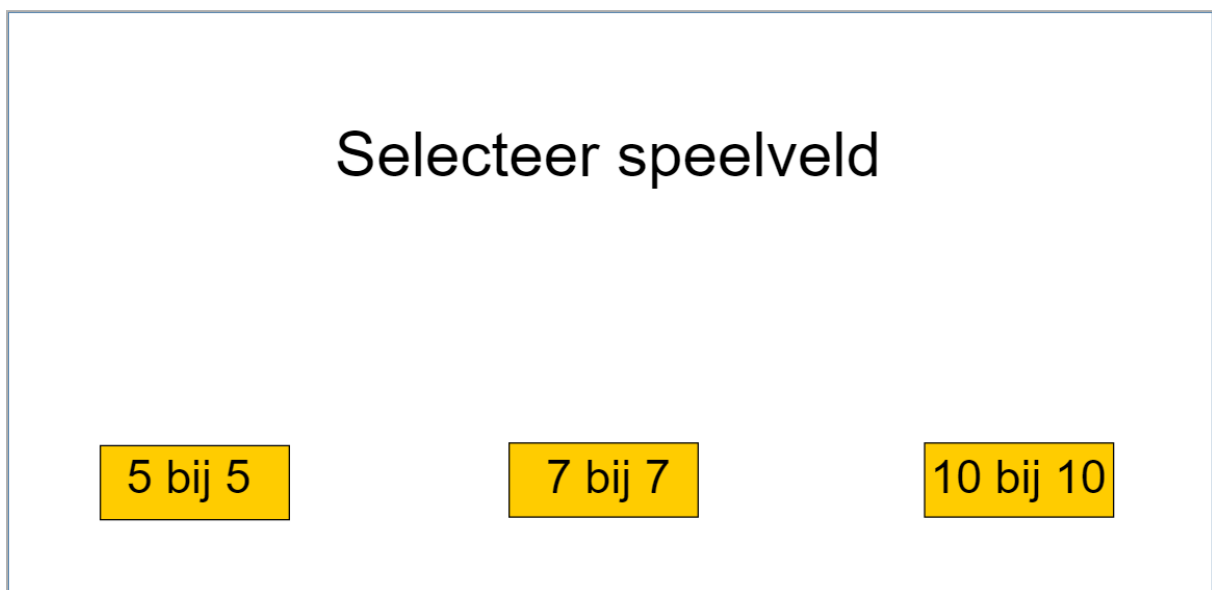
#### Startscherm

Op het startscherm staat de naam van het spel en er staan twee knoppen een start knop, die brengt je naar de volgende pagina, wat gamemode scherm is. Er staat ook een stop knop die zorgt er voor dat het spel zich afsluit.



#### Gamemode scherm

Op dit scherm kan je kiezen hoe groot speelveld je wilt. Je hebt 3 keuzes verdeeld onder 3 knoppen. De keuzes zijn: 5 bij 5, 7 bij 7 en 10 bij 10. Als je deze knop indrukt wordt de waarde van de knop opgeslagen zodat bij het veld tekenen deze waarden kan worden gebruikt. Zodat je altijd het aantal vakjes hebt die zijn geselecteerd.



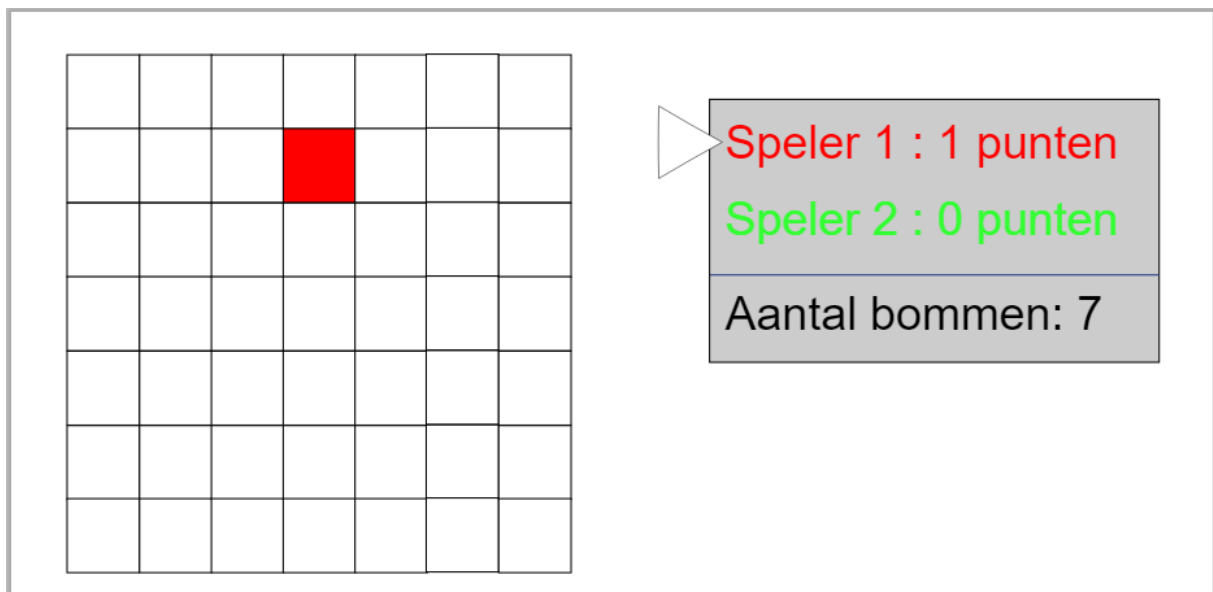
### Levelschermb

Op dit scherm vind het spel plaats, links staat het speelveld(deze verschilt dus door te kijken welke mode is geselecteerd) en rechts staat het dashboard. Hier staan de spelers op, welke kleuren ze hebben en hun aantal punten dat ze hebben. Ook staat er nog onderaan het dashboard aantal bommen. Dit is hoeveel bommen er nog in het veld zijn. Ook zal er als de eerste beurt begint een pijl worden getoond bij de speler die aan de beurt is. Er wordt nu random gekozen welke speler er mag beginnen.



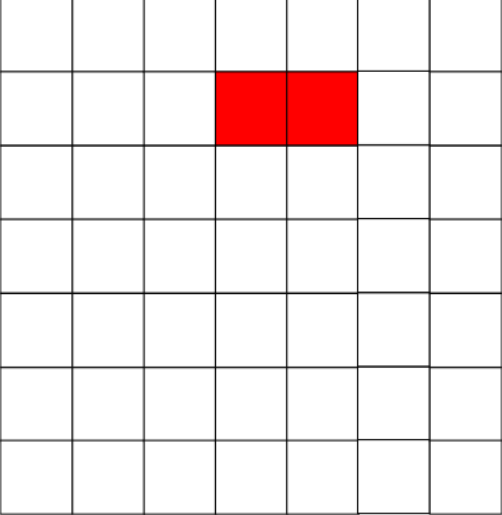
### Eerste beurt

Er is random gekozen dat speler 1 mocht beginnen. Er komt een pijl bij speler 1 te staan en het aantal bommen is ingesteld. Nu kan speler 1 overal klikken op het scherm waar hij wilt. Er is 10% kans dat het een bom is, dan gaat de beurt over naar de volgende speler. Maar speler 1 gokt goed en er zit geen bom achter. Het vakje wordt dan ingekleurd in zijn kleur en hij krijgt een punt.



### Hogere scores maken

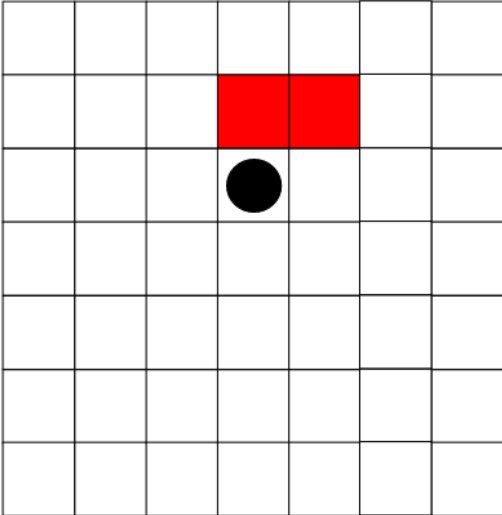
Als je naast, boven of onder al een blokje hebt ingekleurd krijg je een punt extra. In dit voorbeeld is dat de linker naast het blokje dat je indrukt, deze is al ingekleurd. Dit betekent dus dat je in plaats van 1 punt in totaal 2 punten krijgt. Je kan ook meerdere naast, boven of onder je al hebben ingekleurd. Stel dat de blokjes boven en onder je al zijn ingekleurd krijg je dus 3 punten in plaats van 1.



Speler 1 : 3 punten  
Speler 2 : 0 punten  
Aantal bommen: 7

### Bom raken

Als je een blokje raakt met een bom er onder is je beurt over en gaat de beurt naar de andere speler. Dit wordt aangegeven met de pijl die verschuift naar de andere speler. Ook wordt er -1 gedaan van het aantal bommen want er is een bom uit het spel. Ook krijgt de speler die op de bom klikt een minpunt van zijn score.



Speler 1 : 2 punten  
Speler 2 : 0 punten  
Aantal bommen: 6

### Eindscherm

Als alle blokken zonder bommen zijn geclaimd is het spel af. Op het eindscherm wordt bekend gemaakt wie er heeft gewonnen. Je hebt gewonnen door het meeste aantal punten te halen. Er wordt op dit scherm ook weergegeven met hoeveel meer punten de speler die heeft zegen gevierd heeft gewonnen. Op dit scherm staan 2 buttons, de button stoppen sluit het programma af en de knop opnieuw brengt je terug naar het gamemode scherm.





## 4. Softwarebeschrijving

In dit hoofdstuk beschrijf ik op een technische manier hoe het programma moet gaan werken. Dit werk ik uit door te vertellen welke events ik wil gaan gebruiken, welke methodes ik ga gebruiken en welke variabele ik ga gebruiken.

### Globale software structuur

De volgende modules ga ik gebruiken met het schrijven van mijn code:

- De algemene module: hier zal de programma setup draaien, hier zal ook worden ingesteld dat het beeld volledig scherm is en hier worden alle globale variabele gedeclareerd.
- De teken module: hierin zal alles wat vaker moet worden getekend worden opgeslagen, bijvoorbeeld buttons, ook tekent de teken module het speelveld.
- De functies module: hier staan functies in die meerdere malen worden gebruikt, bijvoorbeeld wat er moet gebeuren als een speler dood is. Dit gebeurt namelijk meestal meerdere malen per spel
- De spelmodule: hierin wordt alles gezet wat nodig is om het speelveld te laten werken. Denk bijvoorbeeld aan het overzicht van de aantal punten.

### Events, methode en variabelen

Hieronder staan opgeschreven welke events er van belang zijn voor het programma en het programma op zal reageren ook staan er variabele en methode die het programma moet gebruiken.

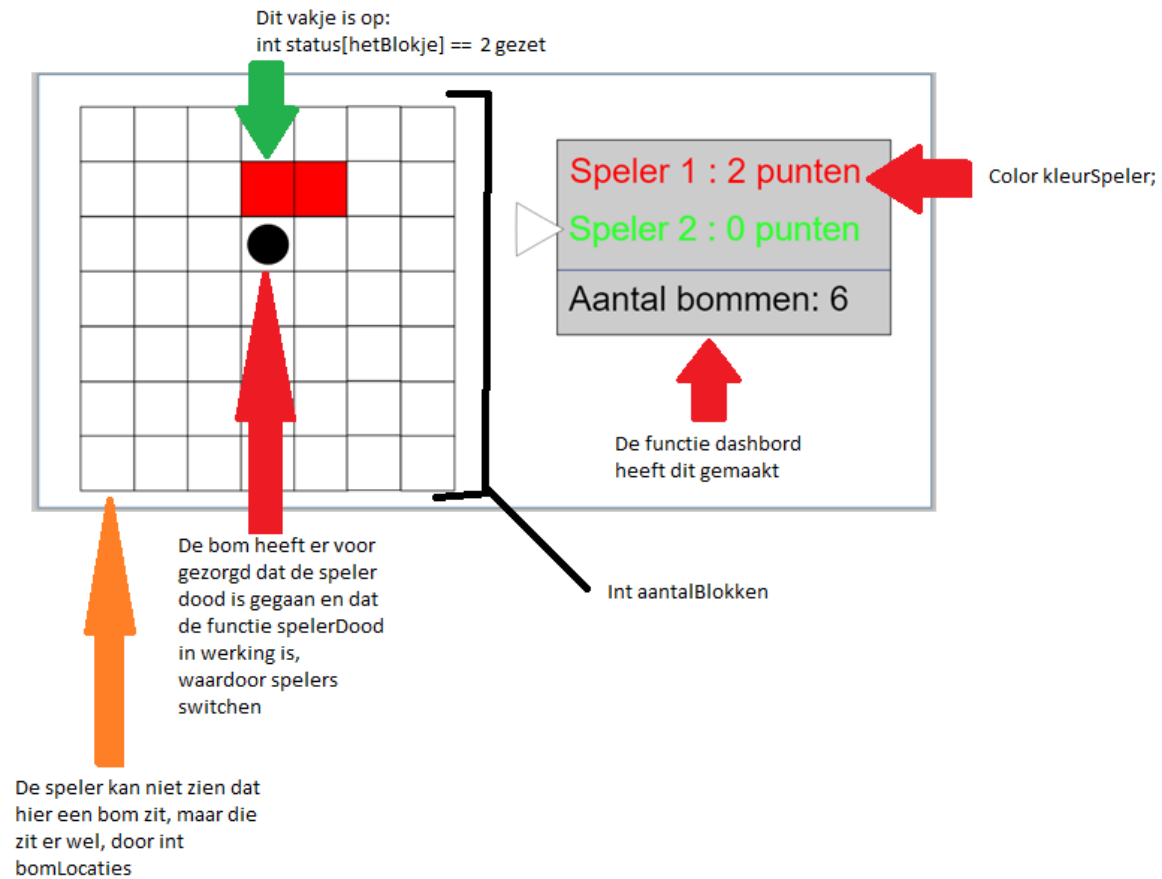
Module	Event/Methode/Variabele	Toelichting
<u>Algemeen</u>	fullscreen()	Voor het gebruiken van het beeld op volledige grote
	Int status[]	Geeft aan of het veld al bezet is, of dat er een bom onder licht.
	Int speelveld[][]	Hiermee maak je het speelveld en zorgt er voor dat elk vakje ook makkelijk als knop kan worden gebruikt voor het spel
	Boolean startscherm, gamemodescherm, levelscherm, eindscherm	Hiermee schakelt de game over tussen de verschillende soorten schermen. Dit doet de game door het scherm wat nu is op false te zetten en het nieuwe scherm op true te zetten.
	mouseClicked	Hierin worden alle dingen gezet die moeten gebeuren als de muis wordt ingedrukt.
<u>Tekenen</u>	void tekenButton	Hiermee worden de buttons getekend, het enige wat je moet doen is een x en een y waarden invullen in de functies.
	void tekenTekstVak	Dit tekent een vierkant met daarin tekst, dit is handig als je dit op meerderen schermen wilt gebruiken

	Void tekenSpeelveld	Tekent de vakjes van het spel, zodat spelers zien waar ze kunnen klikken om te beginnen met spelen.
.	Void tekenBlok	Dit is voor als een persoon op een button heeft geklikt en deze verandert naar de kleur van de speler en deze knop niet nog een keer kan worden ingedrukt.
	Void Dashbord	Hiermee teken je het dashboard met de data van de speler die aan zet is en welke niet. Hoeveel punten elke speler heeft en welke kleur ze hebben. Ook kan je het aantal resterende bommen zien.
<u>Functies</u>	Boolean eindspel	Deze boolean controleert of het spel al afgelopen is en als dat zo is return hij de waarden dat de boolean= true.
	Boolean opVlak	Met deze functie kan worden gecontroleerd of als er een mouseClicked komt of de muis op het veld is van de button, als dit waar is return hij de waarde dat de boolean waarden true is.
	Color kleurSpeler	Hiermee wordt random geselecteerd welke kleur elke speler krijgt, de kleuren zijn wel zo gekozen dat ze niet teveel op elkaar lijken.
	Int bomLocaties	Hier staat in vastgesteld wat de bomlocaties worden.
	Void spelerDood	Als een speler dood gaat door een bom treedt deze functie in werking, hierdoor wordt de speler gewisseld.
	Void beginnendeSpeler	Hiermee wordt vooraf van de game bepaald welke speler er mag beginnen. Dit wordt random gedaan.
<u>Spel</u>	Void speelveldMaken	Dit wordt als voorbereiding gedaan voor het spel, hier wordt alles klaar gemaakt om te spelen.
	Void muisGekliktSpel	Hier staat alles in wat er moet gebeuren als de muis wordt ingedrukt.
	Int aantalBlokken	Hier in wordt aangegeven met hoeveel blokken die moet gaan tekenen. 5 betekent 5 bij 5 enzovoort.

### Samenhang van de events, methoden en variabelen met de ontwerpen

Hieronder staat in een tekening van een schermontwerp, hier staat met pijlen bij waar je kan zien wat iets doet in het ontwerp.

Deze variabelen, methoden en event komen overeen met de variabelen, methoden en events in de tabel hier boven en zijn ter verduidelijking voor het ontwerp



### De programma flow

Hier onder staat een globale opzet van het programma. De programma flow is voorzien van pseudo code en is natuurlijk niet helemaal compleet:

```
Setup(){
    fullscreen();
    Startscherm = true;
}
Draw(){
    if (startscherm){
        tekenButton(rood, "Stop");
        tekenButton(groen, "Start");
        tekenTekstVak(grijs, "Word 2 Conquer");
    }
    if (gamemodescherm){
        tekenButton(oranje, "5 bij 5");
        tekenButton(oranje, "7 bij 7");
        tekenButton(oranje, "10 bij 10");
        tekenTekstVak(grijs, "Kies je gamemode:");
    }
    if (levelscherm){
        Dashboard();
    }
    if(eindscherm){
        tekenButton(rood, "Stop");
        tekenButton(groen, "Nogmaals");
        if(P1 > P2){ tekenTekstVak(grijs, "Speler 1 heeft gewonnen met: " + P1 - P2 + " Punten");
        if(P2 > P1){ tekenTekstVak(grijs, "Speler 2 heeft gewonnen met: " + P2 - P1 + " Punten");
        }
    }
}
mouseClicked()
if (startscherm) {
    if (opVlak(button1)) {
        startscherm = false;
        exit();
    }
    if (opVlak(button2)) {
        gamemodescherm = true;
        startscherm = false;
    }
}
if (gamemodeScherm) {
    if (opVlak(button1)) {
        gamemodescherm = false;
        levelscherm = true;
        aantalBlokken = 5;
        speelveldMaken();
    }
    if (opVlak(button2)) {
        gamemodescherm = false;
        levelscherm = true;
        aantalBlokken = 7;
        speelveldMaken();
    }
}
```

```
    }
    if (opVlak(button1)) {
        gamemodescher = false;
        levelscher = true;
        aantalBlokken = 5;
        speelveldMaken();

    }
}
if(levelscher){
    muisGekliktSpel();
    if(eindspel){
        levelscher = false;
        eindscher = true;
    }
}
if(eindscher){
    if (opVlak(button1)) {
        eindscher = false;
        exit();
    }
    if (opVlak(button2)) {
        gamemodescher = true;
        eindscher = false;
    }
}
}

Void speelveldMaken(){
    for (int i = 0; i < speelveld.length; i++) {
        status[i] = 0;
        rect(speelveld[i][0], speelveld[i][1], eenVakje, eenVakje);
    }
}

Void muisGekliktSpel(){
    if (speelveldX && speelveldY{
        veldenBezeten++;
        if (player1) {
            scoreP1++;
        }
        if (player2) {
            scoreP2++;
        }
    }
}
```

## 5. Testcases

Hieronder staan 4 verschillende testcases, voor elk scherm heb ik gekozen om 1 testcase te maken. Zo kan je goed testen of elk scherm helemaal werkt.

### Testcase 1: kiezen tussen spel stoppen of starten

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT
1.1	Klik op de stop button	Exit();	Het programma stopt
1.2	Klik op de start knop	Startscherm = false; Gamemodescherm=true;	Hij gaat naar het scherm waar je gamemodes kan selecteren
1.3	Gebruik het kruisje van het programma		Het programma stopt
1.4	Geen keuze maken		Er gebeurt niks

### Testcase 2: kiezen tussen gamemodes

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT
2.1	Klik op de 5 bij 5 knop	Gamemodescherm = false; levelschermblokken=true; aantalBlokken = 5;	Hij gaat naar het volgende scherm met een spel van 5 bij 5
2.2	Klik op de 7 bij 7 knop	Gamemodescherm = false; levelschermblokken=true; aantalBlokken = 7;	Hij gaat naar het volgende scherm met een spel van 7 bij 7
2.3	Klik op de 10 bij 10 knop	Gamemodescherm = false; levelschermblokken=true; aantalBlokken = 10;	Hij gaat naar het volgende scherm met een spel van 10 bij 10
2.4	Geen keuze maken		Er gebeurt niks
2.5	Gebruik het kruisje van het programma		Het programma stopt

### Testcase 3: het spel spelen

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT
3.1	Klik op een knop als speler 1(en het is geen bom)	Het blokje word van de speler en de speler krijgt een punt	Het blokje krijgt de kleur van de speler, en er komen extra punten bij de puntenstand.
3.2	Klik op een knop als speler 2(en het is geen bom)	Het blokje word van de speler en de speler krijgt een punt	Het blokje krijgt de kleur van de speler, en er komen extra punten bij de puntenstand.
3.3	Klik op een knop als speler 1 en het is een bom	Er wordt een punt afgehaald van de score van de speler en de andere speler is aan de beurt.	Het vakje krijgt een bom figuur op zich en de pijl gaat naar de andere speler toe om aan te geven dat hij aan de beurt is.
3.4	Klik op een knop als speler 2 en het is een bom	Er wordt een punt afgehaald van de score van de speler en de andere speler is aan de beurt.	Het vakje krijgt een bom figuur op zich en de pijl gaat naar de andere speler toe om aan te geven dat hij aan de beurt is.
3.5	Het laatste beschikbare vakje zonder bom wordt aangeklikt	Er wordt een punt opgeteld bij de speler die op het vak klikt Levelschermb = false; Eindschermb = true;	Je komt op het eindschermb uit.
3.6	Geen keuze maken		Er gebeurt niks
3.7	Gebruik het kruisje van het programma		Het programma stopt

### Testcase 4: kiezen tussen spel stoppen of opnieuw spelen

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT
4.1	Klik op de stop button	Exit();	Het programma stopt
4.2	Klik op de opnieuw knop	eindschermb = false; Gamemodeschermb=true;	De game gaat naar het scherm waar je gamemodes kan selecteren
4.3	Gebruik het kruisje van het programma		Het programma stopt
4.4	Geen keuze maken		Er gebeurt niks