Ultimate Subset

Door Sijmen Huizenga

Versie alpha 3

Inhoud

[Inleiding 3](#_Toc401217878)

[Set 4](#_Toc401217879)

[Hoe werkt de game? 4](#_Toc401217880)

[Eigenschappen? 4](#_Toc401217881)

[Wat is een set? 4](#_Toc401217882)

[Functionaliteit 5](#_Toc401217883)

[Menu 5](#_Toc401217884)

[Game 5](#_Toc401217885)

[Scorebord 6](#_Toc401217886)

[About 6](#_Toc401217887)

[Sneltoetsen 6](#_Toc401217888)

[Systeem 7](#_Toc401217889)

[Componenten 7](#_Toc401217890)

[Schermen 7](#_Toc401217891)

[Knoppen 7](#_Toc401217892)

[Game 8](#_Toc401217893)

[Testen 9](#_Toc401217894)

[Functies en Variable 10](#_Toc401217895)

[Bijlage 14](#_Toc401217896)

[Schermontwerp menu 14](#_Toc401217897)

[Schermontwerp game 15](#_Toc401217898)

[Schermontwerp scoreboard 16](#_Toc401217899)

[Schermontwerp about 17](#_Toc401217900)

# Inleiding

In dit rapport is een plan gemaakt voor het programma Ultimate Subset. Er zijn vier onderdelen. Ten eerste is er een uitleg over de game SET. Daarna is in het hoofdstuk analyse alle requirements beschreven. Hier is dus alle functionaliteit van het programma te vinden. Daarna is er een ontwerp met alle structuren, variabele en methode. Dit hoofdstuk is vooral van belang voor programmeurs. Als laatste is er een testplan waarin wordt beschreven hoe na ontwikkeling het programma de functionaliteit getest kan worden.

# Set

## Hoe werkt de game?

In beeld heb je een aantal kaarten. Dit aantal kaarten varieert tussen 9 en 12 afhankelijk of je het originele of een versimpelde verzie speelt. De bedoeling is om drie kaarten te selecteren die een ‘set’ vormen. Als deze kaarten een set vormen, worden deze kaarten van het bord verwijdert en komen er nieuwe kaarten uit de stapel.

De stapel bevat in het begin van het spel alle mogelijke kaarten(27 of 81). Er worden steeds een aantal op tafel gelegd. Vervolgens pakt de speler een set totdat er geen sets meer op tafel liggen of totdat de stapel leeg is. Elke keer als de speler een kaart pakt, wordt het puntenaantal met 1 verhoogd. Het aantal punten is de score van de speler. Dit gaat door totdat alle kaarten op zijn of totdat er geen set meer mogelijk is.

## Eigenschappen?

Elke kaart heeft een aantal eigenschappen. In de verkleinde versie zijn er 3 eigenschappen:

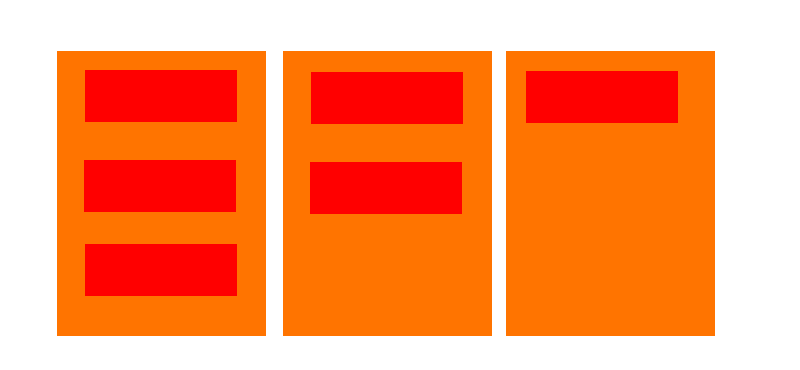
* De hoeveelheid (1, 2, 3)
* De vorm (rechthoek, driehoek, ovaal)
* De kleur van de vorm (rood, geel, blauw)

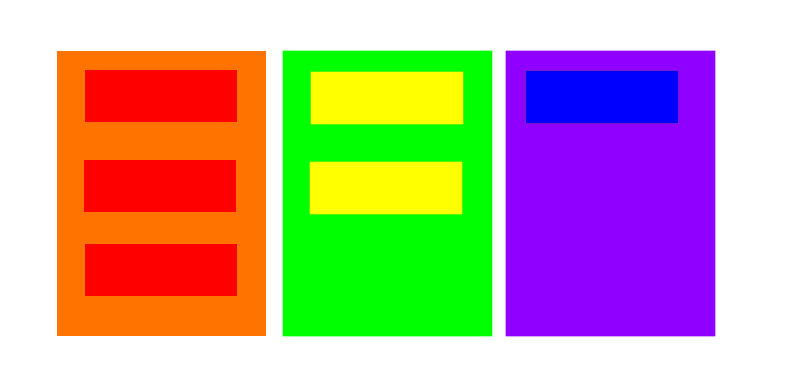
In de originele versie is er nog een extra eigenschap:

* De kleur van de kaart (oranje, groen, paars)

Alle mogelijke kaarten zijn online te vinden: [27 kaarten](https://github.com/SijmenHuizenga/SubSet/blob/master/afbeeldingen/27Kaarten.png) en [81 kaarten.](https://github.com/SijmenHuizenga/SubSet/blob/master/afbeeldingen/81Kaarten.png)

## Wat is een set?

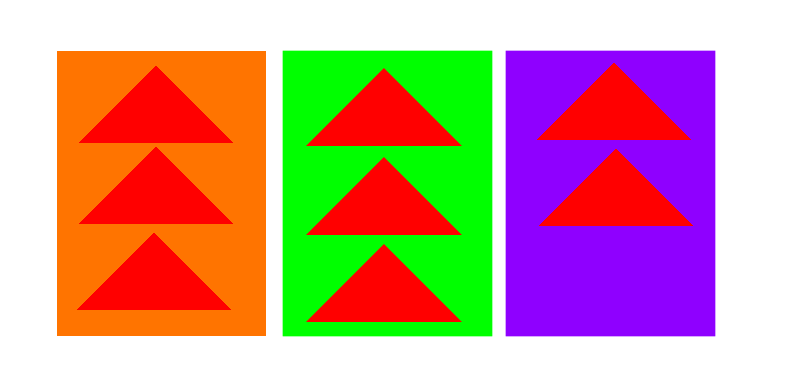
Een set heeft 3 kaarten. Een set kenmerkt zicht dat binnen elke eigenschappen alles verschillend is, of alles anders is. Hier zijn wat voorbeelden:

**voorbeeld 1**  
kleur: rood, rood, rood (overal zelfde)   
achtergrond kleur: oranje, oranje, oranje (overal zelfde)   
vorm: rechthoek, rechthoek, rechthoek (overal zelfde)   
hoeveelheid: 3, 2, 1 (overal anders)  
Dus dit is een valide set.

Voorbeeld

**voorbeeld 2**  
Kleur: rood, geel, blauw (overal anders)   
Achtergrond kleur: oranje, groen, paars (overal anders)   
Vorm: rechthoek, rechthoek, rechthoek (overal gelijk)   
Hoeveelheid: 3, 2, 1 (overal anders)   
Dus dit is een valide set.

Voorbeeld

**voorbeeld 3**  
kleur: rood, rood, rood (overal anders)  
achtergrond kleur: oranje, groen, paars (overal anders)  
vorm: driehoek, driehoek, driehoek (overal gelijk)  
hoeveelheid: 3, 3, 2 (*deels gelijk, deels anders*)  
Dus dit *niet* een valide set.

voorbeeld

# Functionaliteit

Zoals te zien is in de scherm ontwerpen(zie Bijlage), komen er vier schermen. Menu, score bord, about en game. In deze vier hoofdstukken leg ik uit wat er binnen deze schermpjes gebeurt. Daarna is er een opsomming te vinden van alle sneltoetsen.

## Menu

Het menu is het beginscherm van het programma. Vanuit hier kan de speler een aantal verschillende acties ondernemen. Ten eerste kan de speler via de twee witte knoppen het spel starten. Door op een van deze knoppen te drukken word het ‘Game’ scherm geopend. Bij ‘Start Easy Mode’ wordt de game op makkelijke modus gestart, en bij ‘Sart Original Mode’ wordt de game op originele moeilijkheidsgraad gestart. Nog voordat de game gestart is, wordt een invoervakje weergegeven om de naam van de speler te noteren. Deze naam wordt aan het einde samen met de score in het score bord opgeslagen.

Door op de knop ‘ScoreBoard’ te klikken wordt het score bord geopend. Het scherm ‘Scoreboard’ wordt dan dus gestart. Door op de knop ‘About and Rules’ te klikken wordt het scherm ‘About’ weergegeven.

Als laatste is er nog de knop ‘Load saved Game’. Door op deze knop te drukken wordt deze opgeslagen game gestart. Je komt dan uit in het Game scherm. Als er geen opgeslagen game is wordt een error weergegeven.

## Game

Het game scherm heeft een aantal onderdelen. Ten eerste zijn er statistieken. Deze zijn te vinden linksboven in het scherm. Deze statistieken geven bijvoorbeeld de huidige tijd, aantal kaarten op de stapel en de highscore weer.

Daarnaast heb je rechts het speelvlak. Hier liggen 9 of 14 kaarten, afhankelijk of je simpel of original speelt. Door met je muis te klikken en te slepen, kan je de kaarten van locatie veranderen. Op deze manier kan je de kaarten op een volgorde leggen die voor jou makkelijk is.

Als je denkt dat je een set hebt gevonden, dan kan je de kaarten een voor een aanklikken. Dan springen ze naar de vakken linksonder in het scherm. Als je een verkeerde kaart heb aangeklikt, kan je linksonder op de kaart klikken, en dan springt hij weer terug naar het scherm. Als er drie kaarten liggen, dan verschijnt er onder de drie hokjes een knop. Deze knop kan verschillende tekst hebben:  
Zijn de kaarten een set? Dan zegt de knop ‘Set! Hand in’.  
Zijn de kaarten geen set? Dan zegt de knop ‘No set.’  
In het eerste geval kan je de kaarten inleveren, en worden er drie nieuwe kaarten in het speelveld gelegd. Bij het tweede geval kan je op de knop drukken om de geselecteerde kaarten terug te leggen in het speelveld.

Verder zijn er nog vier knoppen die een actie uitvoren. Van boven naar beneden:  
‘**Save & Quit**’: Sla het spelletje op en ga terug naar het menu. Er kan altijd maar één spel zijn opgeslagen. Als er al een spel is opgeslagen wordt deze verwijdert.  
‘**Order Cards**’: Alle kaarten in het speelveld worden op volgorde gelegd zodat het speelveld weer overzichtelijk wordt.  
‘**Hint**’: Als de vakjes linksonder leeg zijn, dan worden er twee kaarten vanuit het speelveld gepakt en in de vakjes linksonder gelegd. Dit kost de speler één minuut.   
‘**Give Up**’: De speler geeft op. De game wordt afgesloten, de score wordt niet genoteerd sinds het spel niet is afgerond.

Op een gegeven moment is de stapel op en zijn alle set’s gevonden: Het spel is afgelopen. Er wordt een popup schermpje weergegeven met de mededeling dat het spel is afgelopen. De score wordt samen met de naam van de speler genoteerd op het scorebord. Op dit moment komt er in beeld de tekst ‘Game Over’ te staan. Met een klik ergens op het scherm wordt het menu weergegeven.

## Scorebord

Op het scorebord wordt de top vijf beste scores weergegeven. Er is een aparte ranking voor Easy en Original mode. Hoe dit er uit ziet is goed te zien in de schermontwerpen(Schermontwerp scoreboard). Verder zijn er nog twee knoppen. De eerste genaamd ‘Clear Scores’ verwijdert alle score lijsten. De tweede knop ‘Back to Menu’ verwijst terug naar het menu scherm.

## About

Op deze pagina is te vinden hoe het spel werkt, hoe je het spel moet spelen, welke dingen van belang zijn enzovoort. De enige knop die hier aanwezig is de knop ‘Back to Menu’, deze knop verwijst naar het menu.

# Software ontwerp

In dit hoofdstuk wordt vertelt hoe de software technisch in elkaar zit. Als eerste wordt de globale structuur van het programma uitgelegd. Daarna is een lijst te vinden van alle functies en daarbij wat deze functie doen.

## Globale structuur

Als je in de source code kijkt, zie je een aantal verschillende bestanden. Deze bestanden stellen allemaal een verschillend onderdeel van het programma weer. In is een tabel te vinden met de verschillende onderdelen en een uitleg over wat ze doen.

|  |  |
| --- | --- |
| Buttons | Alles van knoppen tekenen tot het selecteren van de juiste actie bij een knop wordt binnen dit gedeelte gedaan. |
| Cards | Dit onderdeel handelt alle acties met kaarten. Dus het checken of een aantal kaarten een set zijn en het tekenen van een kaart op het scherm wordt allemaal hier geregeld. |
| Drawing | Het tekenen van het juiste scherm op het juiste moment wordt in dit onderdeel geregeld. Ook valt hier het tekenen van het menu en about scherm onder omdat deze twee geen eigen onderdeel hebben. |
| Game | Dit is zijn alle acties de te maken heb met de game mechanisme. Bijvoorbeeld de game starten en het tekenen van de game valt hier onder. Ook het opslaan en laden van games wordt in dit onderdeel geregeld. |
| Navigation | Dit is een klein onderdeel dat navigatie tussen verschillende schermen regelt. |
| Scores | Binnen dit onderdeel worden alle score lijsten bijgehouden. Hier kan bijvoorbeeld een nieuwe score entry worden toegevoegd. Ook kan hier scorelijsten worden opgeslagen en geladen. |
| Selection | Dit onderdeel zorgt voor de selectie van kaarten binnen de game scherm. Dit onderdeel werkt nou samen met het ‘cards’ onderdeel. |
| Timer | Hier zit de timer die de speeltijd bijhoud. |
| Util | Dit zijn een aantal nuttige functies die niet bij een van de bovenstaande onderdelen horen. |

## Bestandtype

Er worden op twee plekken in het programma data opgeslagen. Hiervoor zijn ook twee bestandstype voor nodig.

**.game**time:12312312354454  
gameType:0  
wrongSets:5  
name:Sijmen  
cardsOnScreen:RON1;RON2;RON3;BQN1;RQN2;RQN3;PTY1;PTY2;PTY3cardsInStack:RON1;RON2;RON3;BQN1;RQN2;RQN3;PTY1;PTY2;PTY3

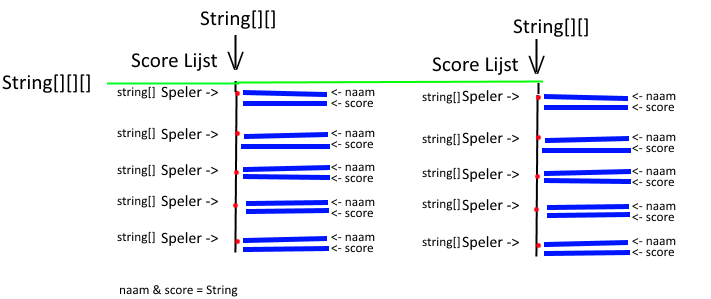
Dit is het bestandstype dat een game opslaat. Het enige nieuwe is de time: dit is de begin tijd in seconden tot nu toe gespeeld. Verder staat gameType 0 voor simple mode, en 1 voor Original mode.

**.scores**  
Sijmen;0:40;Bram;2:50;David;3:00;Max;4:20;Ramon;5:50  
Sijmen;2:40;Bram;2:50;David;3:00;Max;4:20;Ramon;5:50

In een .scores bestand staan altijd twee regels. Elke regel bevat een even aantal stukjes. Deze stukjes zijn gescheiden door een puntkomma. Dit betekent dat de tekst die je opslaat nooit een puntkomma mag bevatten. Je kan steeds groepjes maken van twee stukjes. In het eerste stukje staat een naam, en in het tweede stukje staat een score. In het derde stukje staat weer een naam en in het vierde stukje weer een score.

Elke regel in het bestandje staat één score lijstje. Bij dit programma is de eerste regel de scores van de simpele versie, en de tweede regels is de scores van de originele versie.

## Scorebord

Het scorebord wordt opgeslagen in een drie dimensionaal array met tekst. In figuur 1 is te zien hoe dit array is ingedeeld.  


Figuur

Om met het score bord te werken, zijn er een aantal functies. Ten eerste heb je de *initScoreBoard* functie. Deze moet aan het begin van het programma worden aangeroepen. Binnen deze functie wordt gekeken of het score bestand bestaat. Als het bestand bestaat wordt hij geladen, als hij niet bestaat wordt een nieuw scorebord aangemaakt.

Het laden van scores wordt gedaan door de functie *loadScoreBoard*. Deze heeft als argument een String die de bestandslocatie voorstelt. Deze functie geeft een drie dimensionaal String array terug met daarin alle scores.

Als tegenhanger van *loadScoreBoard* is er *saveScoreBoard*. Deze functie slaat de scores op in een bestand op de computer. Als argument heeft deze functie een String die de bestandslocatie voorstelt en een drie dimensionaal array met de scores die opgeslagen moeten worden.

Om de scores in de goede volgorde weer te geven, moeten de scores geordend worden. Dit gebeurt in de functie *orderScoreBoard*. Deze functie neemt als argument een drie dimensionaal array. Dit array stelt zoals eerder in dit hoofdstuk is uitgelegd de score lijsten voor. Dit array wordt op goede volgorde gelegd.

|  |  |
| --- | --- |
| String[][] list | De score lijst die getekend moet worden. |
| Int x | De x locatie van de lijst. |
| Int y | De y locatie van de lijst. |
| Int wid | De breedte van de lijst |
| Int size | De tekst grote van de tekst binnen de lijst. |

Als laatste is er de *drawScoreList* functie. Hierin wordt een score lijst getekend. Er is een hele lijst met argumenten. Deze zijn weergegeven in de tabel. Deze functie gaat er van uit dat de gegeven lijst al geordend is. Ook is het zo dat nummer 1 een sterretje voor de naam krijgt. Dit het zelfde sterretje wat wordt gebruikt bij buttons met sterretjes.

# Testen

Om uit te vinden of het programma goed werkt, moet er getest worden. Hieronder zijn alle testcases weergegeven.

|  |  |  |
| --- | --- | --- |
| **addButton** | | |
| **Invoer** | **Hoe te testen?** | **Verwachte Uitvoer** |
| “test1”, 3, 2, 50, 100, 150, 200, 255, 15 | Afdrukken van buttonData[][] en buttonTxt in console | buttonData = {{3, 2, 50, 100, 150, 200, 255, 15}}  buttonTxt = {“test1”} |
| De buttonData[][] zit vol. | buttonData[][] vullen en dan de functie addButton aanroepen. | In de console komt de tekst “Not enough space in button data array!” staan. |

# Functies en Variable

String[][] buttonData = new buttonData[buttonAmount][8];

Tabel

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Screen | X | Y | Width | Height | Background Color | Font Color |
| 1 | SCREEN\_MENU | 100 | 300 | 300 | 200 | # FFFFFF | # 00 |
| 2 | SCREEN\_MENU | 500 | 300 | 300 | 200 | # FFFFFF | # 00 |
| 101 | SCREEN\_MENU | 200 | 600 | 500 | 150 | # A00000 | # FFFAFF |
| NULL |  |  |  |  |  |  |  |

Als een row binnen buttonData null is, dan moet deze row genegeerd worden.

String[] buttonText = new String[buttonAmount];

void drawScreen(){  
 swich(selectedScreen){  
 case: SCREEN\_MENU: drawMenu(); break;  
 case: SCREEN\_ GAME: drawGame(); break;  
 case: SCREEN\_ SCORES: drawScores(); break;  
 case: SCREEN\_ ABOUT: drawAbout(); break;  
 }  
}

int getButtonLocation(int id){  
 zoek binnen buttonData naar een button met een identificatienummer wat gelijk is aan id. Als deze gevonden is wordt de locatie terug gegeven. Als hij niet gevonden is wordt -1 terug gegeven.  
}

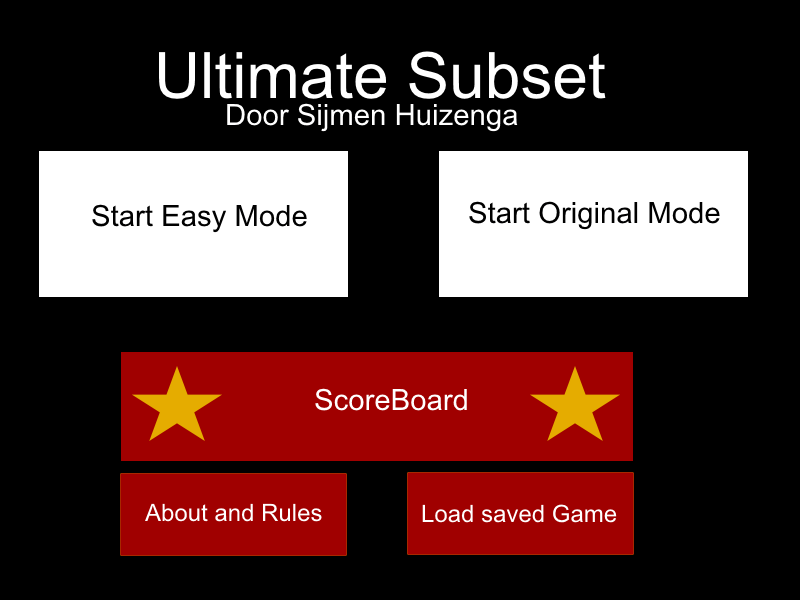
void mouseClicked()  
 for(int[] but : buttonData){  
 if(but[BUTTON\_SCREEN] == selectedScreen){  
 if(mouseX > but[BUTTON\_X] && mouseX < (but[BUTTON\_X]+but[BUTTON\_WIDTH])  
 && mouseY > but[BUTTON\_Y] && mouseY < (but[BUTTON\_Y] + but[BUTOTN\_HEIGHT])){  
 doButtonAction(but[BUTTON\_ID];  
 }  
 }  
}  
}

*doButtonAction(int buttonID){  
 swich(buttonID){  
 case 1: startGame(false); break;  
 case 2: startGame(true); break;  
 case 3: showScoreScreen();break;  
 case 4: showAboutScreen();break;  
 case 5: loadGame();break;  
 case 6: backToMenu();break; //deze staat er twee keer in omdat er twee verschillende   
 case 7: backToMenu();break; // knoppen zijn die deze actie uitvoeren.  
 case 8: clearScores();break;  
 case 9: saveAndQuit();break;  
 case 10: orderCards();break;  
 case 11: hint();break;  
 case 12: giveUp();break;  
 case 12: validInvalidSet();break;  
 }  
 if(buttonID >100 && buttonID <200)  
 cardClickedAction(buttonID);  
}*

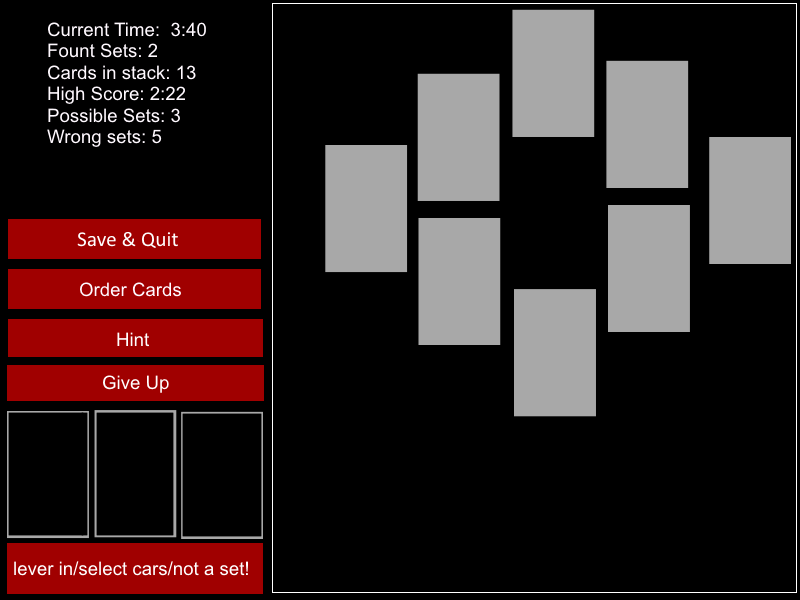
StringList getCardSet(boolean simple){  
StringList out = new StringList();  
 for (int i = 1; i <=3; i++) {  
 for (int j = 1; j <=3; j++) {  
 for (int k = 1; k <=3; k++) {  
 if(simple){  
 out.append(makeCard (i, j, k, 4));  
 }else{  
 for(int l = 1; l <=3; l++){  
 out.append(makeCard (i, j, k, l));  
 }  
 }  
 }  
 }  
 }  
return out;  
}

# Bijlage

## Schermontwerp menu



## Schermontwerp game



## Schermontwerp scoreboard



## Schermontwerp about

