

# Analyse- & ontwerpdocument

BEROEPSPRODUCT – DIVING FOR TREASURE  
ALEX CHENG (634967)

# Inhoud

Inleiding .....	2
Analyse.....	3
Beschrijving van de functionaliteiten .....	3
Schermontwerpen met beschrijving van functionele werking .....	4
Startscherm: .....	4
Uitlegscherm: .....	5
Instellingsscherm: .....	6
Highscorescherm: .....	6
Spelscherm: .....	7
Eindscherm 1 speler:.....	8
Eindscherm 2 speler:.....	9
Ontwerp .....	10
Algoritme in pseudocode .....	10
Globale structuur:.....	11
Main tab – Beroepsproduct_Diving_For_Treasure: .....	11
beginScherm tab:.....	11
uitlegScherm tab:.....	11
settingsScherm tab: .....	11
highscoreScherm tab: .....	11
spelScherm tab: .....	11
eindScherm tab:.....	12
imageSettings tab: .....	12
Te gebruiken methodes en variabelen .....	12
Events.....	12
Globale stappenplan .....	13
Testen .....	14
Test Case Uitlegscherm.....	14
Test Case SettingScherm.....	15
Test Case eindScherm .....	15
Test Case highscoreScherm .....	15
Test Case spelScherm .....	16

# Inleiding

Voor mijn beroepsproduct heb ik het spel Diving for Treasure moeten maken. Volgens de casus speelt dit spel zich af onderwater. Met als scenario een boot waaruit de duikers naar beneden duiken om met als doel munten te verzamelen. Tegelijkertijd liggen er onderwatermijnen die bij aanraking een van de drie levens van de duiker afnemen ook zal er een deel van zijn huidige aantal muntjes afnemen. Behalve muntjes en mijnen moet de duiker ook op zijn zuurstof letten, om zijn zuurstof bij te vullen kan de duiker de zuurstof tanken oppakken die in het water liggen.

De duiker beweegt met zijn toetsenbord knoppen in het water en heeft een speciale 'oppaktoets' om de muntjes in het water op te rapen.

Als de zuurstof op is of de duiker geen levens meer heeft dan eindigt het spel.

Bij het maken van het spel waren een aantal eisen die minimaal erin moesten zitten, ook zijn er optionele eisen die je kunt implementeren als je tijd over hebt. Deze eisen zijn terug te vinden in de casus.

In dit bestand zal ik u vertellen hoe ik het ontwerp, analyse en het testen heb aangepakt tijdens het ontwikkelen van mijn spel, Diving for Treasure.

# Analyse

In het analyse onderdeel van dit document zal doormiddel van een functioneel ontwerp worden verteld wat de opdrachtgever als eindproduct kan verwachten (zonder enige voorkennis te hebben over programmeren/de techniek). Hierbij zullen wij een beschrijving geven van onze functionaliteiten en daarna schetsen/ontwerpen met beschrijving van ons gewenste eindproduct.

## Beschrijving van de functionaliteiten

Zoals er bij de inleiding al werd gezegd heb ik in de casus een aantal minimum eisen gekregen waar ik sowieso aan moest voldoen. Deze waren als volgt:

- Het spel wordt door een speler gespeeld.
- Er zijn zeemijnen en schatten die willekeurig in het scherm stilstaand worden gezet.
- Doormiddel van toetsenbord wordt de besturing gedaan.
- Er zijn drie schermen: startscherm, spelscherm en eindscherm.
- In het startscherm moet je de mogelijkheid krijgen om de hoeveelheid mijnen en munten in te stellen.
- In het eindscherm moet je kunnen aantonen hoeveel geld de uiteindelijke winnaar heeft behaald en de mogelijkheid krijgen om opnieuw te spelen.
- De score blijft constant in beeld.

Daarnaast waren er nog een aantal optionele eisen, de volgende eisen heb ik er uiteindelijk ook bij toegevoegd om het spel een extra uitdaging te geven.

- Het spel is speelbaar voor 1 of 2 spelers.
- 2<sup>e</sup> speler kan besturen met pijlknoppen.
- Alle elementen zijn schaalbaar met het grafische scherm van processing.
- Toevoeging van zuurstof, waardoor je zuurstof tanken moet oppakken om ervoor te zorgen dat je zuurstof niet opraakt.
- Mijnen, munten en zuurstof tanken bewegen.
- Extra scherm toegevoegd waarin het spel kort wordt uitgelegd.

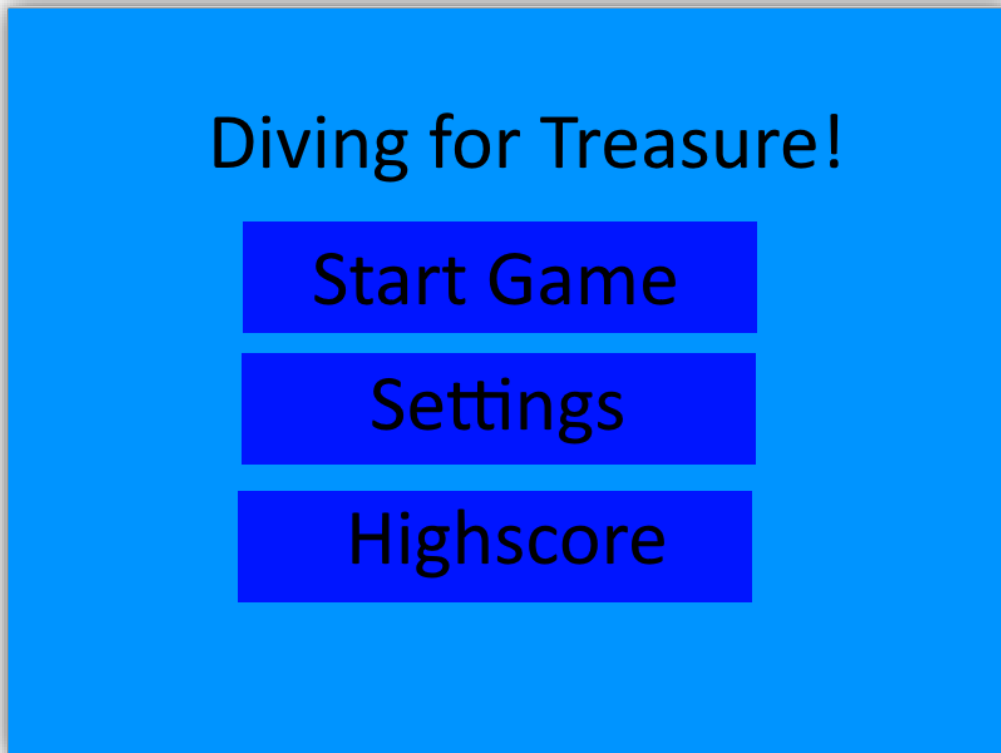
Ten slotte zijn de onderstaande genoemde eisen wel gewenst geweest, maar ben ik er niet aan toegekomen. Dit was of wegens tijdstekort of een gebrek aan kennis over hoe ik het moest aanpakken.

- Een haai die de speler of munten zou kunnen opeten.
- Mogelijkheid om je behaalde highscore op te slaan en later weer te laden.
- Animatie van de duiker, mijnen, munten, boot en haai.
- Meerdere levels waarin de moeilijkheidsgraad hoger is.
- Een haai of 2<sup>e</sup> speler die bestuurd wordt door een AI.

## Schermontwerpen met beschrijving van functionele werking

Volgens de casus moest je minimaal 3 schermen hebben, een startscherm, een spelscherm en een eindscherm. Daarbij heb ik zelf nog een uitleg scherm, een instellingsscherm en een highscorescherm toegevoegd. Mijn schetsen zal ik hieronder aantonen met een beschrijving van de functionele werking. Alle schetsen zijn getekend in PAINT.NET.

Startscherm:



Het startscherm is bedoelt om een eenvoudige look te hebben. Zodat het voor elke gebruiker vriendelijk is. Als je op de 'Start Game' knop klikt ga je verder naar het uitleg scherm. Als je op de 'Settings' knop klikt ga je door naar het instellingsscherm. En als je op de 'Highscore' knopt klikt ga je naar het scherm waarin de hoogste score weergeven worden.

*Achteraf gezien is dit niet gelukt vanwege een gebrek aan kennis hoe ik de highscore save file moest toepassen. Daarom is dit nu de zogenaamde 'Exit Game' knop geworden die het spel weer afsluit.*

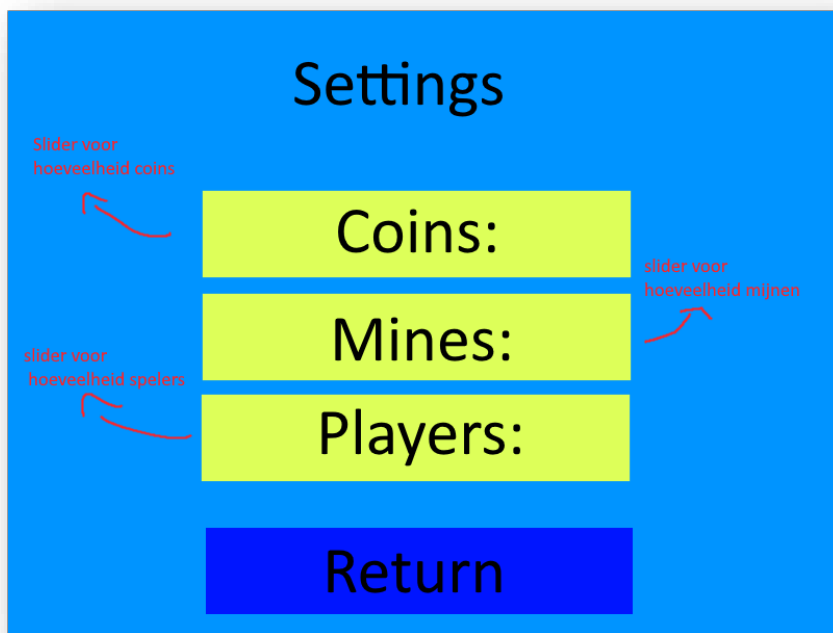
Verder wou ik een achtergrond toepassen die te maken had met het zoeken van schatten onderwater.

Uitlegscherm:



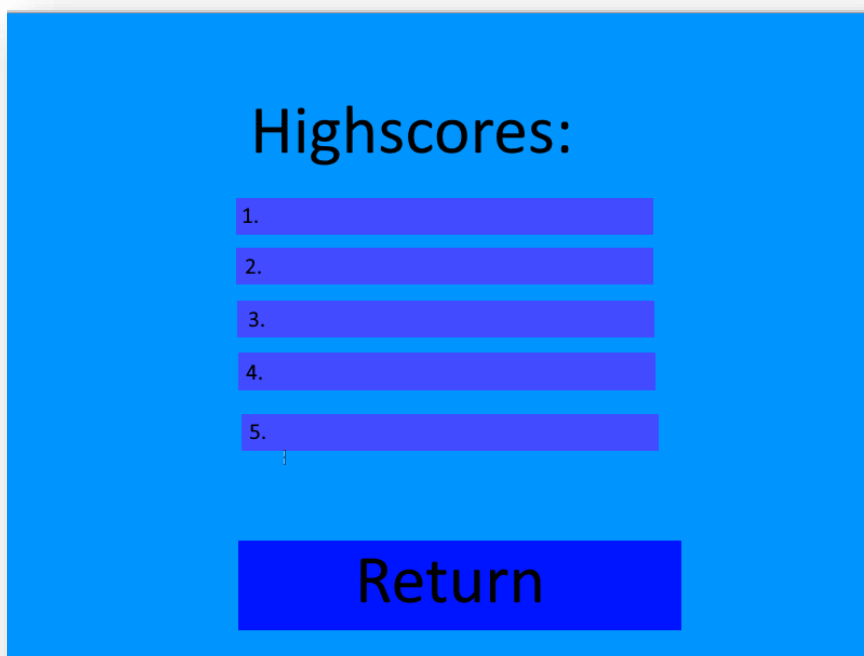
Het uitlegscherm komt nadat je op de 'Start Game' knop klikt. Het enige wat er op dit scherm te zien is een kort maar krachtige uitleg over het spel. Dit houdt in wat het doel van de speler is, de besturingsknoppen, wat de munten & mijnen zijn en hoe je deze hoeveelheid kunt veranderen, hoe het zuurstof systeem werkt en hoe op het eindscherm je jouw score kunt zien / opslaan. Verder is het de bedoeling om dit scherm net zoals het startscherm eenvoudig te houden.

Instellingsscherm:



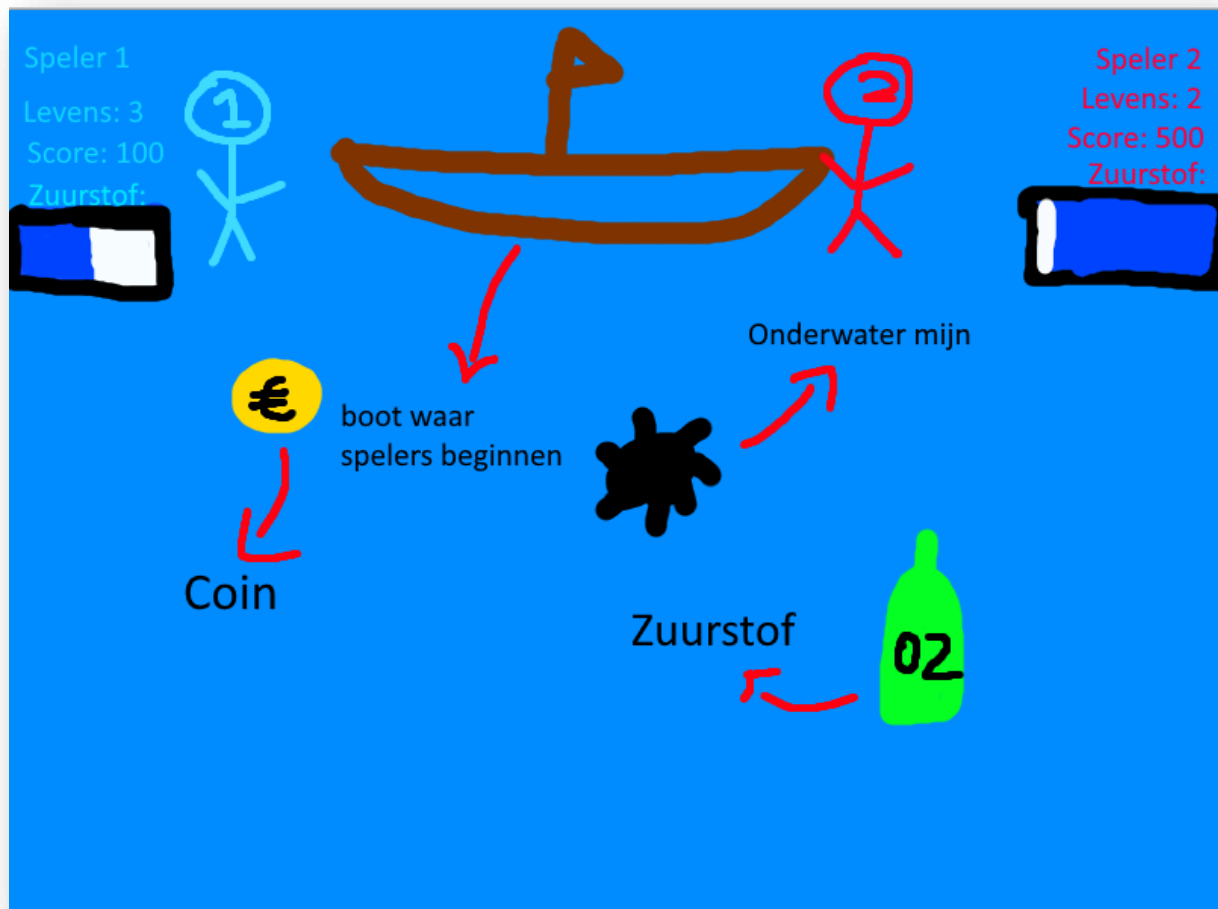
In het instellingsscherm kun je de verschillende waardes van het aantal spelers, munten en mijnen veranderen. Dit wordt doormiddel van verschuifbare schuiven gedaan. Daarnaast kun je weer terug gaan naar het beginscherm door op de 'return' knop te klikken.

Highscorescherm:



Het highscorescherm toont de top 5 hoogst behaalde scores aan. Verder is er nog een 'return' knop om terug naar het beginscherm te gaan.

Spelscherm:



Dit is een van de belangrijkste, of misschien zelfs het belangrijkste scherm. Hier wordt het spel zelf in afgespeeld. Op het scherm zijn veel elementen te zien. Ten eerste de boot waar speler 1 (en eventueel speler 2 zal beginnen). Links en rechts in de hoeken zijn de speler levens, score en zuurstof hoeveelheid te zien. De levens en score zullen simpelweg nummers zijn, terwijl de zuurstof een balk is die langzaam leeg kan lopen.

In het water zullen er dan ook munten liggen die bij aanraking score toe zullen voegen. Behalve muntjes in het water zijn er ook mijnen die bij aanraking een leven weg zullen halen. En ten slotte zuurstof tanken die bij aanraking je zuurstof balk weer aanvullen tot vol.

Verder kunnen de spelers in het scherm van links naar rechts en van boven naar beneden bewegen. Ze kunnen niet buiten het scherm gaan.

Als je geen levens meer over hebt of je zuurstof balk leeg is, ga je naar het eind scherm.

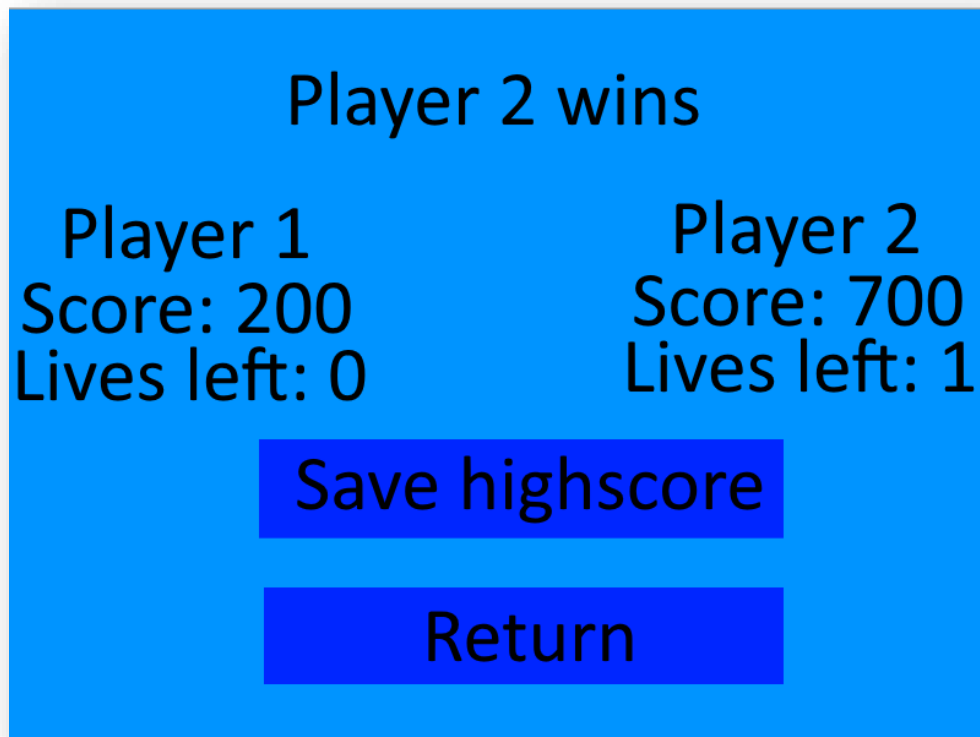


Eindscherm 1 speler:



Zo zou het eindscherm eruit zien als er maar 1 speler geselecteerd is in het instellingsscherm. Je uiteindelijke score gebaseerd op de opgepikte munten en de tijd overleefd worden op het scherm aangetoond. Verder zijn er twee knoppen, 'return' knop die brengt je terug naar het beginscherm om de mogelijkheid te geven om weer te spelen. En een 'save highscore' knop waarmee je jouw behaalde score kunt opslaan en later weer kunt bekijken in het highscorescherm.

Eindscherm 2 speler:



Als er in het instellingsscherm 2 spelers waren gekozen zou het eindscherm er iets anders uitzien. Namelijk zou er eerst een winnaar worden bepaald gebaseerd op de behaalde score. Daarnaast worden de scores en levens over van de twee spelers getoond, om voor we spelers de mogelijkheid te geven om hun score na afloop te bekijken. Daarnaast zijn er net zoals op de eindscherm voor 1 speler weer twee knoppen met de zelfde functie.

# Ontwerp

Bij het ontwerp onderdeel gaan we richting de technische kant. Hierin laat ik zo duidelijk mogelijk zijn wat mijn plan van aanpak is en op hoe ik het ga programmeren. Ook zal ik in een planning aantonen op welke volgorde ik de onderdelen van mijn spel zal ontwikkelen.

## Algoritme in pseudocode

- Eerst zal er doormiddel van een gameState variabele bepaald worden waar het spel zich op dit moment bevindt.
- We beginnen bij gameState = 0, oftewel het beginscherm. Dit beginscherm bevat drie knoppen die bij een mousePressed detectie zullen veranderen naar andere gameStates / schermen.
- Eerste knop is "Start Game" knop die gameState op 1 klikt en van scherm veranderd.
- Als gameState op 1 is veranderd het scherm naar het uitleg scherm. Hierin staat simpel weg uitleg in text met daarna nog een knop die het spel zal starten. En dus weer gameState zal gaan veranderen.
- Tweede knop is op start scherm is "Settings" knop. Deze zal gameState op 2 zetten en naar instellingsscherm gaan. Dit scherm bevat 3 sliders waarmee je de maxCoins, maxMines en maxPlayers mee kunt aanpassen. (Deze zul je verder nodig hebben in het spel zelf) Dit scherm bevat een knop die terug zal gaan het hoofd menu, en dus gameState weer terug naar 0 zet.
- Als laatste is er een highscore knop die naar het highscore scherm zal gaan. Deze laadt doormiddel van save files de 5 hoogst behaalde highscores in. Ook hier is er weer een knop die terug zal gaan naar het hoofd menu.
- Terug naar het spelscherm, gameState zal op 5 worden gezet. Hier zullen er gebaseerd op de maxCoins & maxMines die ingesteld zijn in het instellingsscherm op willekeurige plekken munten en mijnen op het scherm neergezet worden. Ook zullen er levens, de zuurstof tijd en munten worden bijgehouden met een score, lives, oxygenTime functie.
- Als er een tweede speler geselecteerd is moet er een extra lives, levens en zuurstof tijd bijgehouden worden. (denk aan lives2, score2, oxygenTime2)
- Ook zal de beweging per speler bijgehouden moet worden. Eerst moet doormiddel van key en keyCode een keyPressed & keyReleased detecteert worden. Daarna moet de bewegingssnelheid opgeteld of afgetrokken worden van de huidige X/Y positie van de speler om een beweging te stimuleren.
- Daarnaast moet er een object aanraking gedetecteerd kunnen worden tussen de speler en de mijnen, munten en zuurstoftanken.
- Het spel let erop wanneer de zuurstof of levens op zijn, als dit gebeurt wordt de gameState op 6 gezet en gaat het spel meteen door naar het eindscherm
- Het eindscherm leest de score/levens/tijd af van het spelscherm en toont deze aan op het eindscherm.
- Hier is dan een 'highscore' knop die ervoor zorgt dat je jouw behaalde score kunt opslaan (en kunt laden vanaf het 'highscore' knop op het beginscherm)
- Ten slotte is hier een 'return' knop die terug zal gaan naar het beginscherm (gameState 0).

## Globale structuur:

Ik ben van plan om verschillende tabbladen te maken gebaseerd op het aantal schermen in het spel.

Main tab – Beroepsproduct\_Diving\_For\_Treasure:

Dit tabblad wordt gewoon vernoemd naar hoe de map gaat heten om het openen van het spel makkelijk te houden. Hierin staat de void setup en void draw. In de setup staat de grootte van het scherm en in de void draw de methode die de gameState zal bijhouden.

GameState tab:

In het gameState tabblad zal ik de verschillende gameStates bijhouden die het wisselen van schermen bijhoudt. Verder komt er hier de methode die in de void draw staat met daarin alle verschillende schermen. Ook zal hier de mousePressed staan om de knoppen bij te houden voor scherm veranderingen.

beginScherm tab:

Hierin zal alles in staan wat met het beginscherm temaken heeft. De eerste regel van deze methode is dan ook meteen `if(gameState == 0){`, dus hij speelt alles af wat in gameState 0 staat en als het gameState 0 is. Hierin heb ik een methode aangemaakt die de knoppen zal maken, gebaseerd op de parameters. Wat deze knoppen doen wordt in de gameState geregeld.

uitlegScherm tab:

Het uitlegscherm zal weer met de tekenKnoppen methode van beginScherm een knop hebben die verder naar het spel zal gaan. Ook is er verder alleen tekst die het spel kort uitlegt.

settingsScherm tab:

Hierin staan methodes met parameters op de drie verschillende Sliders te maken. Met een methode die de positie van de slider blok `return()`'d. Verder staat er weer de tekenKnop methode erin die ervoor zorgt dat men terug naar het beginScherm kan.

highscoreScherm tab:

In het highscoreScherm worden de savefiles ingeladen en is er weer een tekenKnop om terug te gaan naar het beginScherm.

spelScherm tab:

Hierin wordt het meeste gedaan. Verschillende methodes die doormiddel van een 2<sup>d</sup> array de willekeurige positie bepalen voor de mijnen, munten en zuurstof tanken. Verder worden hier alle toetsenbord bewegingen geregistreerd en meegenomen in een 1d array. Die daarna doorgaat naar de methode om de beweging van de speler te regelen. Ook worden er aparte methodes aangemaakt die de score, levens en zuurstof teller bijhouden. Ten slotte met een methode zal ik dan de teksten aanmaken om het aantal benodigde lines te verminderen.

eindSchermb tab:

Hierin staat een methode die bepaald wie het spel heeft gewonnen als er twee spelers meedoen. Anders wordt er gewoon verteld wat de uiteindelijk score van speler 1 was. Daarnaast zijn er weer met de tekenKnoppen methode twee knoppen getekend. Een voor highscore opslaan en een voor het terugaan naar beginscherm.

imageSettings tab:

Hierin worden alle images geladen en geresized met een imageSettings() methode. Deze staat in de void setup.

## Te gebruiken methodes en variabelen

**gameState()** - houdt huidige gameState bij.

**startSchermb()** - hierin staat alles van het startSchermb.

**tekenKnoppen(float knopNummer, String knopTekst)** - Maakt knoppen aan met keuze voor eigen tekst en knop positie.

**settingsSchermb()** - hierin staat alles van het settingsSchermb.

**tekenSlider(float sliderX, float sliderY, float sliderBreedte, float sliderHoogte, int positie, int nPositie)** - maakt de slider aan voor het kiezen van hoeveelheid muntjes, mijnen, spelers.

**bepaaldPositie(float sliderX, float sliderBreedte, int nPosities)** - Geeft een return() waarde voor positie van sliderblokje.

**spelSchermb()** - hierin staat alles van spelSchermb.

**uitlegSchermb()** – hierin staat alles van uitlegSchermb.

**eindSchermb()** – hierin staat alles van eindSchermb

**balBeweging()** – leest keyPressed/keyReleased af en beweegt hiermee de speler.

**tekenSpeler()** – Plaatst de speler in het scherm om door balBeweging te besturen.

**spawnMuntjes()** – zorgt voor willekeurige plaatsing muntjes.

**spawnVallen()** – zorgt voor willekeurige plaatsing mijnen.

**spawnZuurstof()** – zorgt voor willekeurige plaatsing zuurstof.

**maakItemCoin()** – maakt de willekeurige positie aan voor spawnMuntjes().

**maakItemMine()** - maakt de willekeurige positie aan voor spawnVallen ().

**maakItemZuurstof()** - maakt de willekeurige positie aan voor spawnZuurstof().

**scoreTeller()** – Houdt de score bij van speler 1 & speler 2.

**levenTeller()** – Houdt de levens bij van speler 1 & speler 2.

**zuurstofTeller()** – Houdt de zuurstof bij van speler 1 & speler 2.

**imageSettings()** – Worden alle settings voor images geregeld en samen in setup gezet.

## Events

**mousePressed()** - gebruikt om de gameState veranderingen bij te houden. (Als knop wordt gedrukt)

**keyPressed()** – registreert wanneer toetsenbord voor beweging wordt ingedrukt.

**keyReleased()** – registreert wanneer toetsenbord voor beweging wordt losgelaten.

## Globale stappenplan

Ik ben van plan om het per tabblad aan te pakken en dus mijn project in kleine delen te splitsen. Hierdoor zou het makkelijker worden voor ontdekken van fouten en het oplossen hiervan. Daarna probeer ik zoveel gebruik te maken van methodes om het erg overzichtelijk voor mijzelf en de opdrachtgever te houden. Ook ga ik dan rustig mijn algoritme volgen en het baseren op mijn schermontwerpen van het ontwerp onderdeel.

# Testen

In het laatste onderdeel van mijn analyse- en ontwerpdocument ga ik alles testen. Dit doe ik volgens het test case schema die wij bij SAQ hebben geleerd. Ook zal ik het per scherm doen om het overzichtelijk te houden.

*Omdat uiteindelijk is gebleken dat het highscore savefile te implementeren niet is gelukt heb ik dit ook niet in mijn eindproduct gelaten.*

## Test Case Startscherm

Testcase	Actie	Verwachte Resultaat	Geslaagd?
1	Mousepressed naast alle knoppen.	Gebeurt er niets	Ja
2	Mousepressed op 'start game' knop.	Zet gameState op 3 en gaat door naar uitlegScherm	Ja
3	mousePressed op 'settings' knop.	Zet gameState op 1 en gaat door naar settingsScherm	Ja
4	mousePressed op 'exit game' knop.	Sluit het spel af	Ja
5	mousePressed op 'highscore' knop	Gaat naar highScore scherm en doet gameState op 6	Ja

## Test Case Uitlegscherm

Testcase	Actie	Verwachte Resultaat	Geslaagd?
1	Mousepressed naast knop.	Gebeurt er niets	Ja
2	Scherm grootte verandert	Tekst blijft op schaal als scherm veranderd van grootte	Ja
3	Mousepressed op 'Continue to Game' knop	Zet gameState op 4 en gaat naar spelScherm	Ja

## Test Case SettingScherm

Testcase	Actie	Verwachte Resultaat	Geslaagd?
1	Muis beweging binnen de slider met mousePressed	Beweegt sliderblokje en hoeveelheid tekst	Ja
2	mousePressed naast knop	Gebeurt er niets	Ja
3	mousePressed op 'Save Settings' knop	Zet gameState weer op 0 en terug naar beginScherm	Ja
4	Van beginscherm voor 2 <sup>e</sup> keer terug naar settingScherm	Instellingen nog steeds zelfde als wat ingesteld was	Ja

## Test Case eindScherm

Testcase	Actie	Verwachte Resultaat	Geslaagd?
1	Met 1 speler naar eindscherm	Toont behaalde score	Ja
2	Met 2 spelers naar eindscherm	Bepaald winnaar en toont score per speler	Ja
3	Mousepress naast knop	Doet niets	Ja
4	Mousepressed op 'Return to home screen scherm'	Gaat terug naar beginScherm en reset alle benodigde waardes	Nee, alleen tijd niet correct
5	Verander schermgrootte	Tekst veranderd correct mee	Ja

## Test Case highscoreScherm

Testcase	Actie	Verwachte Resultaat	Geslaagd?
1	Mousepressed naast knop	Niets	Ja
2	Mousepressed op 'Return to homescreen' knop	Terug naar beginscherm en gameState weer op 0	Ja
3	Scherm grootte veranderd	Schaalt mee	Ja
4	Vraagt savefile op	Laat savefile vanuit tekstdocument	Nee
5	Leest scores vanuit array	Sorteer scores van hoog naar laag	Ja



## Test Case SpelScherf

Testcase	Actie	Verwachte Resultaat	Geslaagd?
1	Start spel	Aantal gekozen munten/mijnen/spelers kloppen met wat gekozen was	Ja
2	Uit spelscherf bewegen	Lukt niet en houdt spelers binnen spelscherf	Ja
3	Speler botsing met mijn	Verliest een leven en wordt terug naar boven gestuurd, en mijn verdwijnt	Ja
4	Speler botsing met munt	Score neemt toe en munt verdwijnt, spawn nieuwe en speler begint weer bovenaan	Ja
5	Als munt/mijn/zuurstoftank bovenkant aanraakt	Verdwijnt munt/mijn/zuurstoftank en komt nieuwe	Ja
6	Als zuurstof op is	Naar eindscherf	Ja
7	Als beide spelers zelfde munt/mijn opraken	Beide speler krijgen score/leven aftrek	Nee
8	Als je 3000 millis wacht voor spelscherf (gebruik millis timer voor zuurstof)	Spel gaat direct door naar eindscherf	Nee
9	Als je mijn opraakt met 0 munten	Score gaat naar negatieve waarde	Ja
10	Scherfgroote veranderd	Alle schaal mee	Ja
11	Speler botsing met zuurstoftank	Zuurstoftank tijd wordt weer gereset	Ja