# 1. Functioneel ontwerp

## 1.1 Het spel

Een bepaald aantal munten wordt in een vierkant gelegd, met de kop naar boven. Wanneer een munt wordt aangewezen, moet die munt tezamen met de aangrenzende munten worden omgedraaid. Aangrenzende munten zijn degene die links, rechts, boven of onder de aangewezen munt liggen. Wanneer een munt wordt aangewezen waarvan de aangrenzende munten reeds omgedraaid zijn, worden de aangrenzende munten weer omgedraaid, zodat deze weer met de kop naar boven liggen.

Het spel is afgelopen wanneer alle munten met de kop naar beneden liggen. Het doel is om dit met zo min mogelijk zetten te bereiken.

## 1.2 functionaliteiten

### 1.2.1 Vereist

Het aantal munten waaruit het vierkant bestaat wordt bepaald aan de hand van het gekozen niveau; het spel zal dus meerdere niveaus kennen. Het niveau kan worden geselecteerd uit een overzicht van alle beschikbare niveaus.

Per niveau zal worden bijgehouden wat het laagste aantal zetten is waarmee de speler het niveau heeft voltooid. Deze gegevens kunnen worden ingezien in een overzichtelijk scherm en blijven bewaard wanneer het programma wordt afgesloten, zodat ze opgehaald kunnen worden nadat het programma weer is opgestart.

### 1.2.2 Optioneel

Per niveau is de moeilijkheidsgraad instelbaar. Deze moeilijkheidsgraad wordt bepaald aan de hand van maximaal aantal zetten en/of tijdslimiet.

## 1.3 Schermen

Tabel 1.3 weergeeft per scherm welk doel het scherm heeft, welke functionaliteiten het kent en welke gevolgen deze functionaliteiten hebben.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scherm** | **Beschrijving** | **Functionaliteiten** | |
| **Functionaliteit** | **Gevolg** |
| ***Hoofdmenu*** | Het hoofdmenu is het eerste scherm dat na het starten van het spel getoond wordt. | Knop “Spelen” | Na het aanklikken van deze knop krijgt de gebruiker het scherm *niveauselectie* te zien. |
| Knop “Highscores” | Na het aanklikken van deze knop krijgt de gebruiker het scherm *highscores* te zien. |
| ***Niveauselectie*** | Dit scherm wordt bereikt vanuit het hoofdmenu.  Het doel van dit scherm is om de gebruiker verschillende niveaus te laten selecteren voor het spel | Per niveau een knop “Niveau *X*” | Na het aanklikken van deze knop begint het spel met het corresponderende niveau. Dit speelt zich af in scherm *spel*. |
| Knop “Hoofdmenu” | Na het aanklikken van deze knop krijgt de gebruiker het scherm *hoofdmenu* te zien. |
| ***Highscores*** | Dit scherm wordt bereikt vanuit het hoofdmenu.  Het doel van dit scherm is om de gebruiker per niveau de highscore te tonen. | Klikken binnen het scherm. | Na op een willekeurige positie in het scherm te hebben geklikt, krijgt de gebruiker het scherm *hoofdmenu* te zien. |
| ***Spel*** | Dit scherm wordt bereikt vanuit het niveauselectiescherm.  In dit scherm wordt het spel gespeeld. | Knop “Terug” | Na het aanklikken van deze knop krijgt de gebruiker het scherm *niveauselectie* te zien. |
| Knop “Reset” | Na het aanklikken van deze knop worden alle zetten in het spel ongedaan gemaakt en kan de gebruiker opnieuw beginnen. |
| Voltooien niveau | Als de gebruiker het niveau voltooid heeft, krijgt deze in hetzelfde scherm de huidige score en nieuwe highscore te zien. |

Tabel 1.3

## 1.4 Globale Flow

Vanuit het hoofdmenu is zowel het scherm higshcores als niveauselectie te benaderen. Vanaf deze twee schermen is tevens het hoofdmenu te benaderen. Slechts vanuit het scherm niveauselectie is het spel te starten en vanuit het spel is alleen het scherm niveauselectie te benaderen.

Afbeelding 1.4 weergeeft de globale flow tussen de schermen van de applicatie.

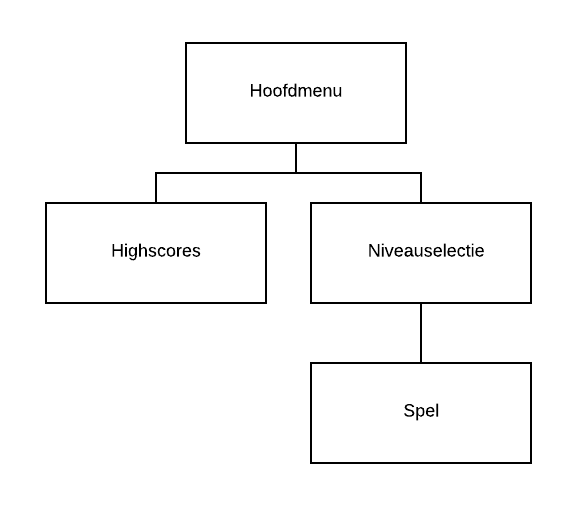
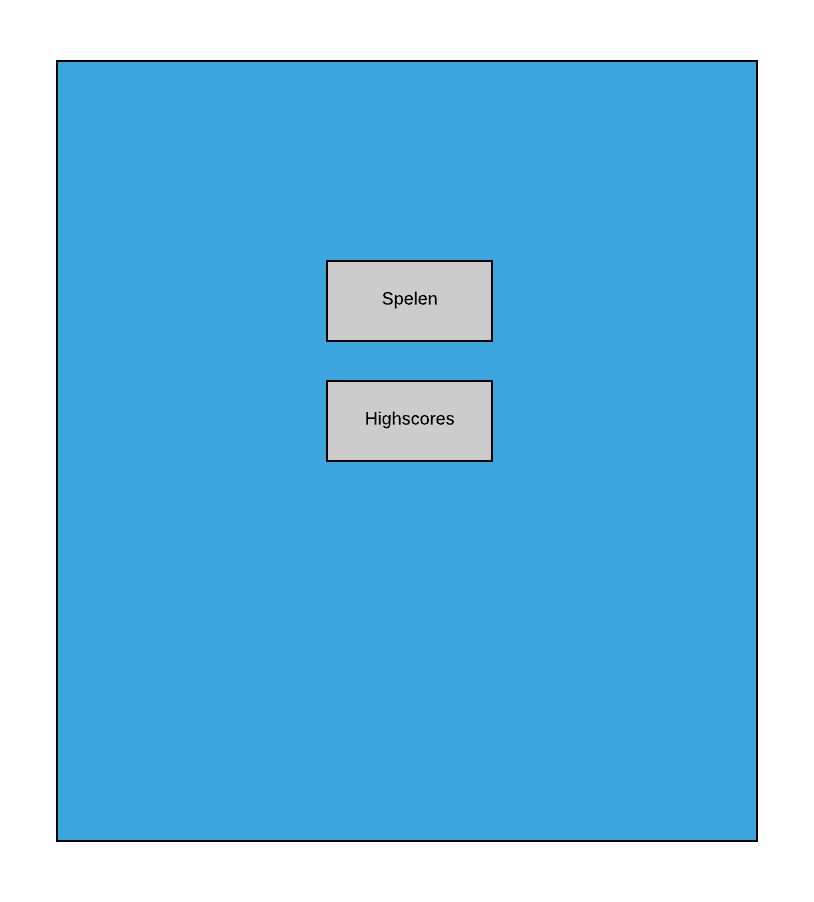


Figure 1.4 Globale flow tussen schermen

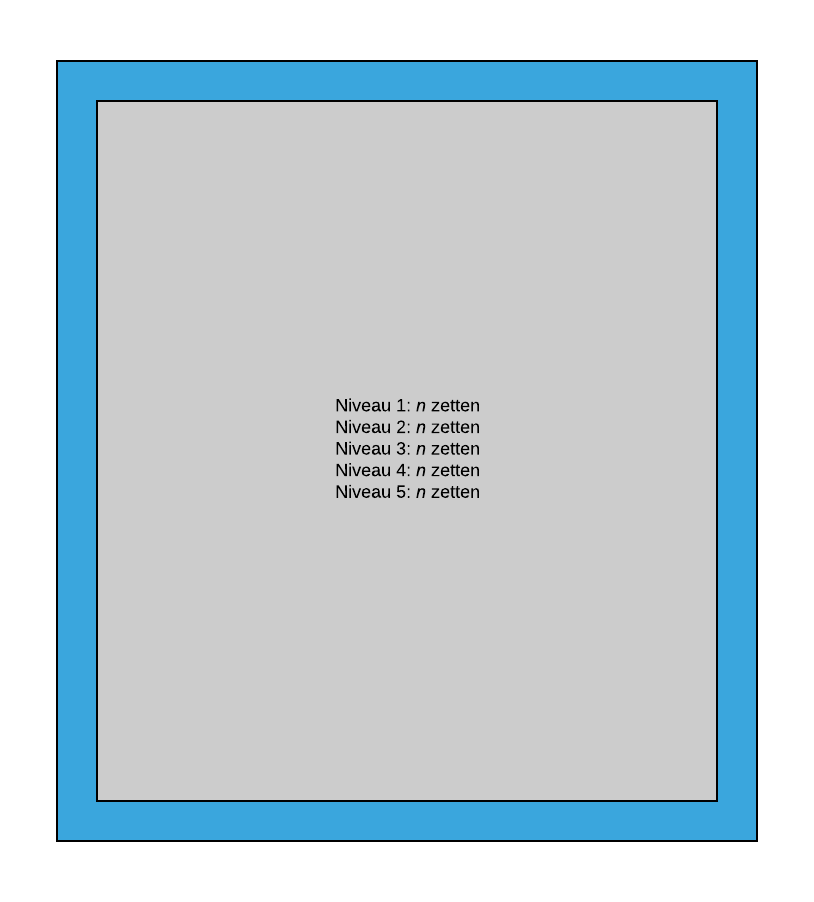
## 1.5 Schermontwerp

In deze paragraaf wordt per scherm van het programma een ontwerp weergeven.

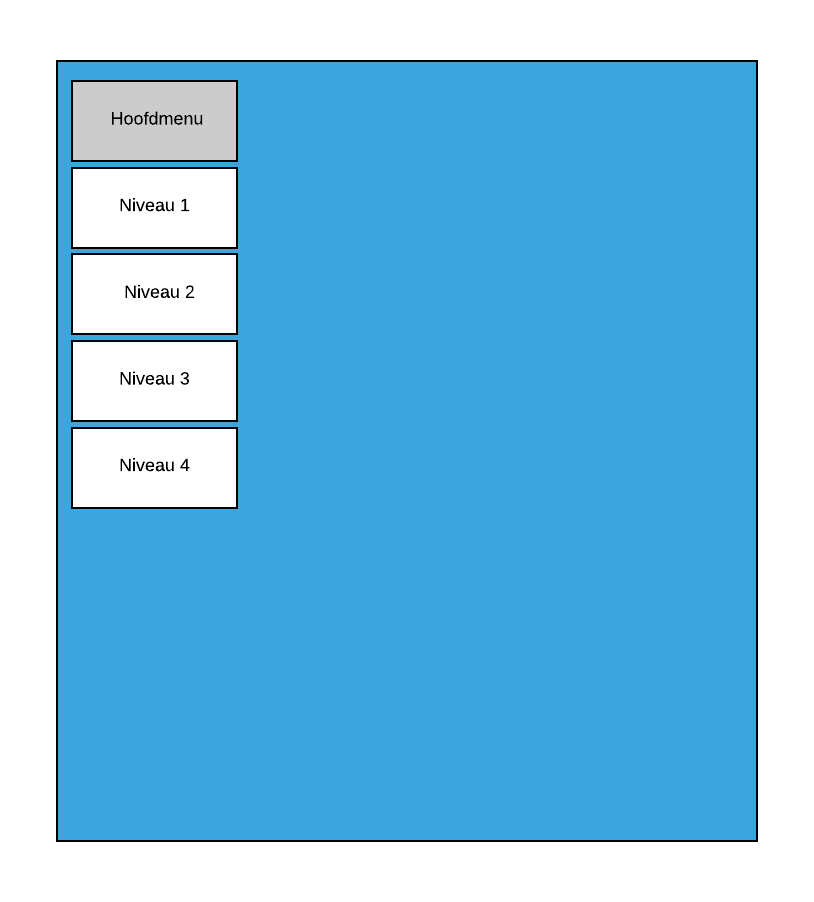
### 1.5.1 Hoofdmenu



### 1.5.2 Highscores



### 1.5.3 Niveauselectie



### 1.5.4 Spel

### https://documents.lucidchart.com/documents/f1d96b90-7f05-4afe-9570-5f8420f4ca4b/pages/0_0?a=522&x=496&y=38&w=847&h=902&store=1&accept=image%2F*&auth=LCA%209a6b234a920315afb0dd42fafbbf9d7ce81df5c3-ts%3D1551204716

# 2. Technisch Ontwerp

(..)

## 2.1 SoftwareBeschrijving

### 2.1.1 Globale Structuur

De software is opgedeeld in modules waarvan alle methodes en variabelen bij hetzelfde doel horen.

Per scherm van het programma is er één module. Tevens is er naast de standaardmodule van een processing programma ook een module *algemeen*, die alle methodes en variabelen bevat die regelmatig door de rest van de modules benut wordt.

In tabel 2.1.1 worden de modules van de software beschreven.

|  |  |  |
| --- | --- | --- |
| **Module** | **Beschrijving** | **Gebruikt door modules** |
| standaardmodule | Dit is de hoofdmodule van de software. In deze module wordt het weergeven van de verschillende schermen aangeroepen en wordt de gebruikersinvoer verwerkt. | N/a |
| Hoofdmenu | Bevat alle instructies met betrekking tot de functionaliteiten van het scherm *hoofdmenu*. | *Algemeen* |
| Niveauselectie | Bevat alle instructies met betrekking tot de functionaliteiten van het scherm *niveauselectie*. | *Algemeen* |
| Highscores | Bevat alle instructies met betrekking tot de functionaliteiten van het scherm *highscores*. | *Algemeen* |
| Spel | Bevat alle instructies met betrekking tot de functionaliteiten van het scherm *spel*. | *Algemeen* |
| Algemeen | Bevat alle functies die gebruikt kunnen worden voor generieke procedures en daarom niet tot slechts één van de bovenstaande modules behoren. | *Hoofdmenu, niveauselectie, highscores, spel* |

Tabel 2.1.1 Globale structuur

### 2.1.2 Program Flow

Deze paragraaf beschrijft de belangrijkste flows binnen het programma.

#### 2.1.2.1 Main Flow

De main flow van het programma begint met het initialiseren van de verschillende scherm en het definiëren van welk scherm als eerst getoond moet worden. Daarna komt de flow in een loop die zich blijft herhalen gedurende de tijd dat het programma actief is. In deze loop worden twee dingen gecontroleerd: moet het scherm geüpdatet worden en/of moet gebruikersinput met de muis worden verwerkt. In het eerste geval wordt de procedure *teken huidig scherm* aangeroepen; in het tweede geval wordt de procedure *verwerk input voor huidig scherm* aangeroepen. De instructies voor deze procedure bevinden zich in de standaardmodule.

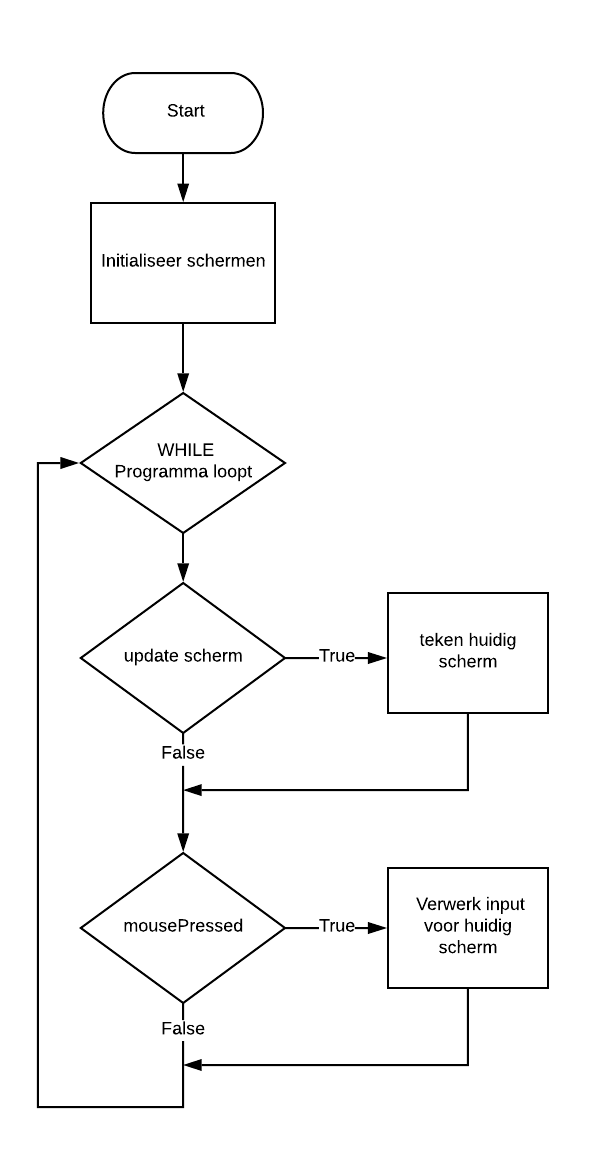


Figure 2.1.2.1 Main flow

#### 2.1.2.2 Teken Huidig Scherm

De procedure voor het tekenen van het huidige scherm wordt aangeroepen door de main flow (§2.1.2.1). Deze procedure vergelijkt met een switch case de numerieke waarde van het huidige scherm met de numerieke waardes van alle mogelijke schermen. Wanneer een overeenkomst gevonden is, wordt de procedure voor het tekenen van het specifieke scherm aangeroepen. De instructies voor deze procedure bevinden zich in de module *algemeen*.

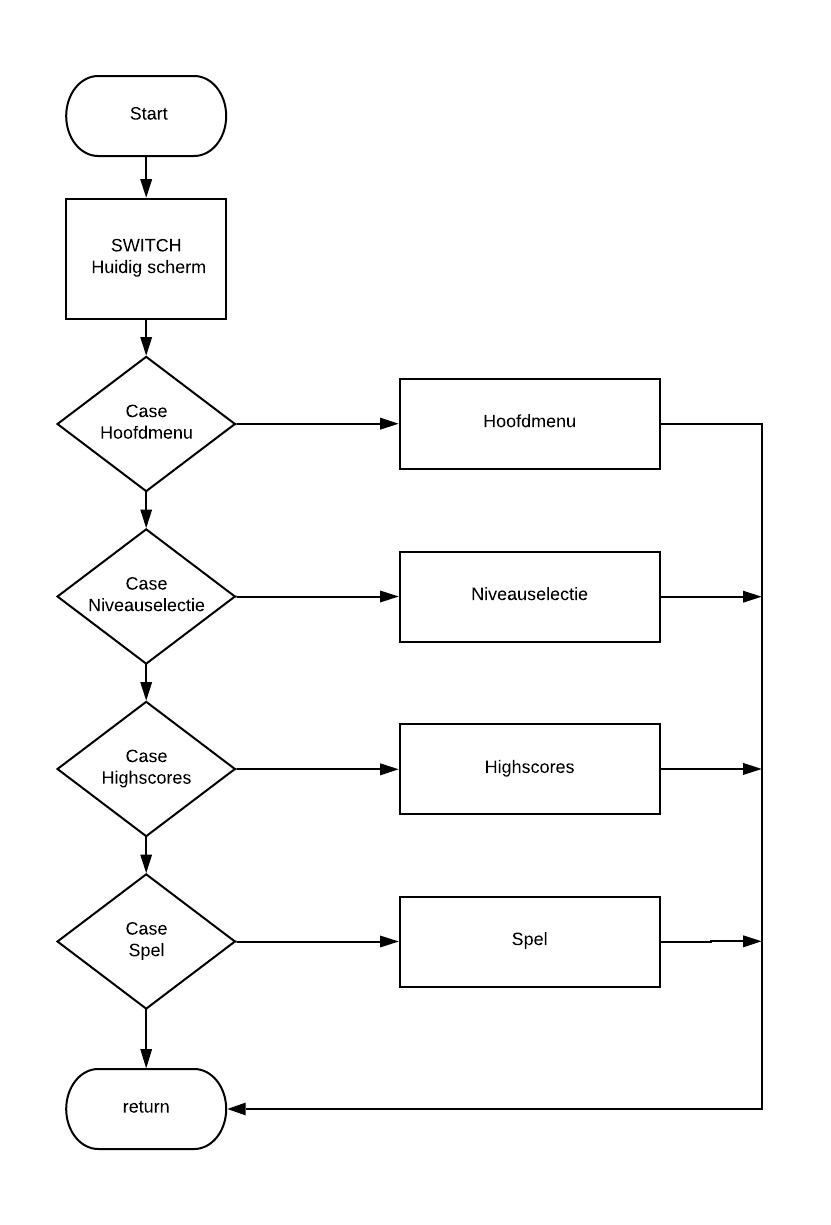


Figure 2.1.2.2 Flow voor “teken huidig scherm”

#### 2.1.2.3 Verwerk Input Voor Huidig Scherm

Deze procedure wordt aangeroepen in de main flow (§2.1.2.1). Net als de procedure “teken huidig scherm” (§2.1.2.2) wordt in deze procedure middels een switch case gecontroleerd wat het huidige scherm is. Wanneer in de switch case een overeenkomst is gevonden, wordt de procedure aangeroepen die voor het betreffende scherm de input verder verwerkt.

In het geval van input voor scherm niveauselectie wordt gecontroleerd of na het verwerken van de input het huidige scherm op *spel* is gezet. In dat geval wordt het scherm voor *spel* geïnitialiseerd.

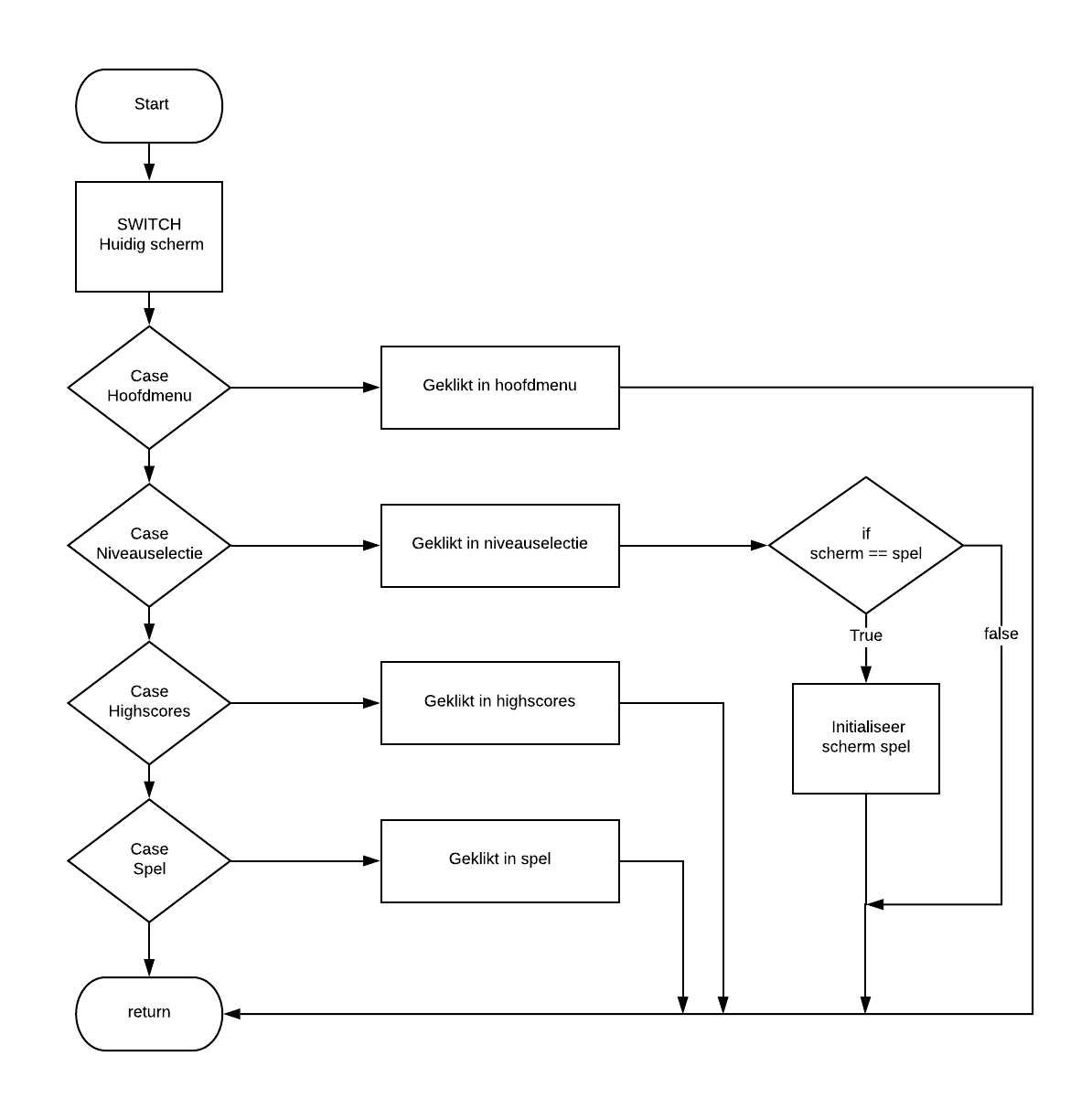


Figure 2.1.3.1 Flow voor “verwerk input voor huidig scherm”

### 2.1.3 Variabelen en Methodes

In tabel 2.1.3.1 worden de namen van de belangrijkste variabelen weergeven, samen met hun beschrijvingen en de modules waarin ze gedeclareerd zijn. Naast deze variabelen zullen er een aantal constantes zijn die de grootte en numerieke waardes van verschillende objecten definiëren.

|  |  |  |
| --- | --- | --- |
| **Variabele** | **Beschrijving** | **Gedeclareerd in module** |
| int scherm | Numerieke waarde van het scherm dat getekend moet worden.  In de program flows “teken huidig scherm” (§2.1.2.2) en “verwerk input voor huidig scherm” (§2.1.2.3) wordt deze variabele gebruikt om het huidige scherm te vergelijken met de numerieke definities van de schermen. | Algemeen |
| int huidigNiveau | Het huidige niveau waarop het spel ingesteld staat. | Algemeen |
| int[] highscores | Array waarin per spelniveau de meest recente highscore wordt bijgehouden. | Algemeen |
| int[][] *x*Knoppen | Waar *x* staat voor de naam van modules niveauselectie, hoofdmenu en spel. Bevat een array van knoppen en hun bijbehorende x- en y coördinaten. De index voor de eerste dimensie zijn de numerieke waardes van de verschillende knoppen. De array in de tweede dimensie heeft twee elementen voor de coördinaten van beide assen. | * Hoofdmenu * niveauselectie * spel |
| int[][][] muntPositie | Een array met drie dimensies voor de x- en y coördinaten van alle munten in het spel. De eerste en tweede dimensies worden geïndexeerd met kolompositie en rijpositie respectievelijk. De derde dimensie bestaat uit een array voor de x- en y coördinaten. | Spel |
| boolean[][] muntOmgedraaid | Een array met twee dimensies voor kolompositie en rijpositie. Een element is *true* als de munt op die positie omgedraaid is, of *false* als dit niet het geval is. | Spel |
| int aantalZetten | Houdt per spelsessie bij hoeveel zetten zijn gezet. | Spel |
| boolean voltooid | Wordt op *true* gezet nadat geconstateerd is dat alle munten in de spelsessie omgedraaid zijn. | Spel |
| int muntenPerZijde | Het aantal munten per zijde van het vierkant. | Spel |

Tabel 2.1.3.1

In paragraaf 2.1.2 zijn al een aantal belangrijke procedures toegelicht. In de software worden deze procedures uitgevoerd door de setup() en draw() methodes van Processing (§2.1.2.1), de tekenScherm() procedure uit de module *algemeen* (§2.1.2.2) en de verwerkInputVoorHuidigScherm() methode uit de module *algemeen* (§2.1.2.3).

De laatste twee methodes roepen methodes aan uit de overige modules, namelijk *hoofdmenu, niveauselectie, highscore* en *spel*. De methodes binnen deze modules hebben elk een doel dat specifiek dient voor het scherm dat correspondeert met de modulenaam.

Tabel 2.1.3.2 licht de belangrijkste algemene methodes die gedeclareerd zijn in de software toe.

|  |  |  |
| --- | --- | --- |
| **Methode** | **Beschrijving** | **Gedeclareerd in module** |
| tekenScherm() | Controleert op basis van variabele *scherm* welk scherm momenteel getekend moet worden en roept op basis daarvan de bijbehorende methode aan. | Algemeen |
| verwerkInputVoorHuidigScherm() | Controleer op basis van variabele *scherm* welk scherm momenteel aan de gebruiker getoond wordt en roept de methode aan die de input voor dat specifieke scherm verwerkt. | Algemeen |
| int gekliktOpKnop( int [][]knoppen,  int x, int y,  int breedte, int hoogte) | Controleert aan de hand van de argumenten of op een knop (en zo ja, op welke) in de array knoppen gedrukt is.  Rapporteert het indexnummer van de knop waarop geklikt is, of -1 als op geen van de knoppen in de array geklikt is. | Algemeen |
| int [][]intialiseerKnoppen(  int n,  int positionering,  int x, int y,  int grootte) | Initialiseert een array voor *n* knoppen en zet de x- en y coördinaten zodanig dat ze naast elkaar (positionering == 0) of onder elkaar (positionering == 1) staan.  Rapporteert een array van *n* knoppen met per knop de gewenste x- en y coördinaten. | Algemeen |
| tekenKnop(  String tekst,  int x, int y,  int breedte, int hoogte,  color knopKleur, color tekstKleur) | Tekent een knop op basis van x- en y positie, hoogte en breedte. Drukt in het midden van de knop de opgegeven tekst af in de opgegeven kleur, met als achtergrond de opgegeven kleur. | Algemeen |
| opslaanHighscores() | Slaat de waardes van de array highscores op in een lokaal tekstbestand “highscores.txt” | Algemeen |
| ophalenHighscores() | Haalt de highscore waardes op uit een lokaal tekstbestand “highscores.txt” en laadt deze in array highscores. | Algemeen |

Tabel 2.1.3.2

Naast deze algemene methodes kent elke module voor een specifiek scherm een aantal standaardmethodes. Deze hebben namen die gebaseerd zijn op het bijbehorende scherm. Tabel 2.1.3.3 weergeeft deze methodes in het algemeen. Daarnaast geeft het ook toelichting over overige methodes die toegelicht moeten worden.

|  |  |  |
| --- | --- | --- |
| **Methode** | **Beschrijving** | **Gedeclareerd in module** |
| Per scherm *x* de functie *x*() | De modules hoofdmenu, niveauselectie, highscores en spel hebben elk een methode waarvan de naam overeenkomt met de naam van het scherm (scherm hoofdmenu kent methode hoofdmenu(), scherm niveauselectie kent methode niveauselectie(), enzovoort.)  Deze methodes bevatten alle instructies die nodig zijn voor het tekenen van de objecten en teksten die het scherm vormgeven volgens het schermontwerp. | * Hoofdmenu * Niveauselectie * Highscores * Spel |
| Per scherm *x* de functie  gekliktIn*X*(int x, int y) | De modules hoofdmenu, niveauselectie, highscores en spel hebben elk een methode “gekliktIn” met daaropvolgend de naam van het scherm. Bijvoorbeeld: gekliktInHoofdmenu(), gekliktInSpel().  Deze methodes bevatten alle instructies die nodig zijn voor het verwerken van muisinvoer voor de gewenste functionaliteiten per scherm. | * Hoofdmenu * Niveauselectie * Highscores * Spel |
| Per scherm *x* de functie *x*Setup() | Initialiseert alle benodigdheden voor de functionaliteiten en het tekenen van objecten van een specifiek scherm. | * Hoofdmenu * Niveauselectie * Spel |
| draaiMuntenOm(  int i, int j) | Draait de munt om op positie i, j. Tevens worden alle aanwezige naastgelegen munten ook omgedraaid. | Spel |
| int[][][] initliaiseerMuntPosities(  int muntenPerZijde, int xStart, int yStart, int muntGrootte) | Zet alle munten in het spel klaar op de juiste posities, zodat ze in het speelveld passen.  Rapporteert een array voor alle munten met hun x- en y coördinaten. | Spel |
| Boolean[][] initialiseerMuntOmgedraaid(  int muntPerZijde) | Zet de waardes van alle munten zodanig dat ze met de kop naar beneden liggen (false).  Rapporteert een array met per munt de waarde voor of deze met de kop omhoog (true) of omlaag (false) ligt. | Spel |
| Boolean alleMuntenOmgedraaid(  Boolean [][]muntOmgedraaid) | Controleert of alle munten in de array muntOmgedraaid met de kop omhoog liggen (true).  Rapporteert true als alle munten met de kop omhoog liggen, of false als dit niet zo is. | Spel |

Tabel 2.1.3.3