SubSet door Sijmen Huizenga

# Systeem

Je kan het programma verdelen in twee onderdelen: Component en Game. Met component bedoel ik alle knoppen, checkboxes, invoervakjes en verschillende schermpjes. Als er bijvoorbeeld een knop op het hoofdmenu staat ‘start game’, en je klikt op deze knop, dan wordt deze klik gehandeld door het component gedeelte van het programma. De game zijn de game mechanieken die het spel verzorgen. Zo houdt de game bijvoorbeeld de punten in de gaten.

## Component

Laten we bij het begin beginnen. Elk scherm(menu, game, scores, about) heeft een nummer. Door middel van dit nummer wordt bijgehouden welk scherm op dit moment wordt weergegeven. Om de schermen aan de nummer te linken, zijn er vier variabele die aangeven welk nummer bij welk scherm hoort:

**int SCREEN\_MENU** = 0; **int SCREEN\_GAME** = 1; **int SCREEN\_SCORES** = 2; **int SCREEN\_ABOUT** = 3;

Om aan te geven welk scherm nu wordt weergegeven gebruik je: **int selectedScreen** = SCREEN\_MENU;

Als dan de het scherm getekend moet worden, wordt naar de ‘selectedScreen’ variabele gekeken en wordt deze gebruikt om de juiste draw functie aan te roepen.  
**swich(selectedScreen){  
 case:** SCREEN\_MENU: drawMenu(); **break;  
 case:** SCREEN\_GAME: drawGame(); **break;  
 case:** SCREEN\_SCORES: drawScores(); **break;  
 case:** SCREEN\_ABOUT: drawAbout(); **break;  
}**

Naast schermen zijn er knoppen. Zoals te zien is in de ‘future screenshots’ zijn er erg veel verschillende knoppen(ongeveer 12). Het is onmogelijk om voor elk van deze knoppen een aparte x, y, breedte en hoogte bij te houden. Daarom gebruik ik en 2D array:

**String[][] buttonData = new buttonData[**aantalButtons**][**8**];**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| 3 | **SCREEN\_MENU** | 200 | 600 | 500 | 150 | # A00000 | # FFFAFF |

Omdat het verwarrend kan zijn welke locatie in het array corresponderen met welke eigenschap, zijn er een aantal variabelen die dit aangeven:

int **BUTTON\_ID** = 0;int **BUTTON\_SCREEN** = 1;int **BUTTON\_X** = 2;int **BUTTON\_Y** = 3;int **BUTTON\_WIDTH** = 4;int **BUTOTN\_HEIGHT** = 5;int **BUTTON\_BGCOLOR** = 6;int **BUTTON\_FGCOLOR** = 7;

Om handig met dit array te kunnen werken is er een functie in het leven geroepen die een button return aan de hand van ID nummer:

**Int[] getButton(int id){  
 for(int[] but :** buttonData**){  
 if(but[**BUTTON\_ID**] == id)  
 return but;  
 }  
 return null;  
}**Zoals te zien is, wordt bij een invalide id NULL teruggegeven.