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

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

final int COL\_RED = color(228, 33, 31);  
final int COL\_BLUE = color(53, 76, 152);  
final int COL\_YELLOW = color(245, 230, 5);  
final int BG\_ORANGE = color(244, 150, 16);  
final int BG\_GREEN = color(79, 169, 45);  
final int BG\_PURPLE = color(107, 55, 138);  
final int BG\_NONE = color(200, 200, 200);

final char C\_COL\_RED = ‘R’;  
final char C\_COL\_BLUE = ‘B’;  
final char C\_COL\_YELLOW = ‘C’  
final char C\_SHAPE\_ELLIPSE = ‘E’;  
final char C\_SHAPE\_QUAD = ‘Q’;  
final char C\_SHAPE\_TRINAGLE = ‘T’;  
final char C\_AMOUNT\_1 = ‘1’;  
final char C\_AMOUNT\_2 = ‘2’;  
final char C\_AMOUNT\_3 = ‘3’;  
final char C\_BG\_ORANGE = ‘O’;  
final char C\_BG\_GREEN = ‘G’;  
final char C\_BG\_PURPLE = ‘P’;  
final char C\_BG\_NONE = ‘N’

int buttonAmount = 3;  
int backgroundColor = 0;  
int selectedScreen = SCREEN\_MENU;  
boolean forceScreenUpdate = true;

String[] cardStack = null;

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

Tabel 1:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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){  
 int i = 0;  
 for(int[] but : buttonData){  
 if(but[BUTTON\_ID] == id)  
 return but;  
 i++;  
 }  
 return -1;  
}

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);  
}*

tabel 2:

|  |  |
| --- | --- |
| **Functienaam** | **Wat doet die?** |
| *startGame* | Start het spel. Een boolean argument geeft aan of subset of set gestart moet worden. |
| *showScoreScreen* | Laat het score bord zien. |
| *showAboutScreen* | Laat het about scherm zien. |
| *loadGame* | Laad een game vanuit de harde schijf. Daarna het gelade spel starten. |
| *backToMenu* | Laat het menu zien. |
| *clearScores* | Verwijder alle scores uit het score menu. |
| *saveAndQuit* | Sla het spel op en sluit het programma |
| *orderCards* | Organizeer de kaarten op het speelveld |
| *hint* | Geef een hint door twee kaarten te selecteren. |
| *giveUp* | Geef op en ga terug naar het menu |
| *validInvalidSet* | Doe een dubele actie. Zie hoofdstuk Functionaliteit paragraaf Game derde alinea. |
| *cardClickedAction* | Als een kaart geklikt is, moet er een speciale actie worden ondernomen. Deze actie krijg dus een aparte functie. Daarom moet ook als parameter het button id worden meegegeven. |

String[] getCardSet(boolean simple){  
String[] out = new String[simple ? 27 : 81];  
 int counter = 0;  
 for (int i = 1; i <=3; i++) {  
 for (int j = 1; j <=3; j++) {  
 for (int k = 1; k <=3; k++) {  
 if(simple){  
 out[counter] = makeCardString(i, j, k, 4);  
 counter++;  
 }else{  
 for(int l = 1; l <=3; l++){  
 out[counter] = makeCardString(i, j, k, l);  
 counter++;  
 }  
 }  
 }  
 }  
 }  
return out;  
}