

M.I.N.T. Projekt: ProgBlock

Inhaltsverzeichnis

M.I.N.T. Projekt: ProgBlock	1
Beschreibung	1
Vorgehensweise	2
Aufbau eines Programms	2
Speichern und Exportieren von Programmen	2
Einführen von neuen Funktionen	2
Aufbauen von Programmen	2
Einen neuen Block hinzufügen	5
Liste der Jetzigen Blöcke	5
Events	5
Actions	6
Data	6
Drawing	6
Operations	7
Pläne des Projekts	8
Fazit	8

Beschreibung

ProgBlock wird ein Programm sein mit dem sich das Programmieren vereinfachen lässt. Es soll Anfängern die Möglichkeit geben das Programmieren mit Hilfe von zusammen-schiebbaren Blöcken zu vereinfachen.

Vorgehensweise

Wir haben ProgBlock mit der programmiersprache Processing geschrieben. Am Anfang des Projekts konnten wir schon teils programmieren mussten jedoch dazu lernen um es zu vollbringen. (<https://www.processing.org/reference>)

Wir fingen damit an die bloße Struktur des Programms zu machen. Als wir fanden, dass dieses funktionsfähig war haben wir dann das Menu und die verschiedenen Blöcke programmiert.

Mit hilfe von diesen Grundlagen haben wir dann angefangen das Endprodukt zu programmieren.

Aufbau eines Programms

Speichern und Exportieren von Programmen

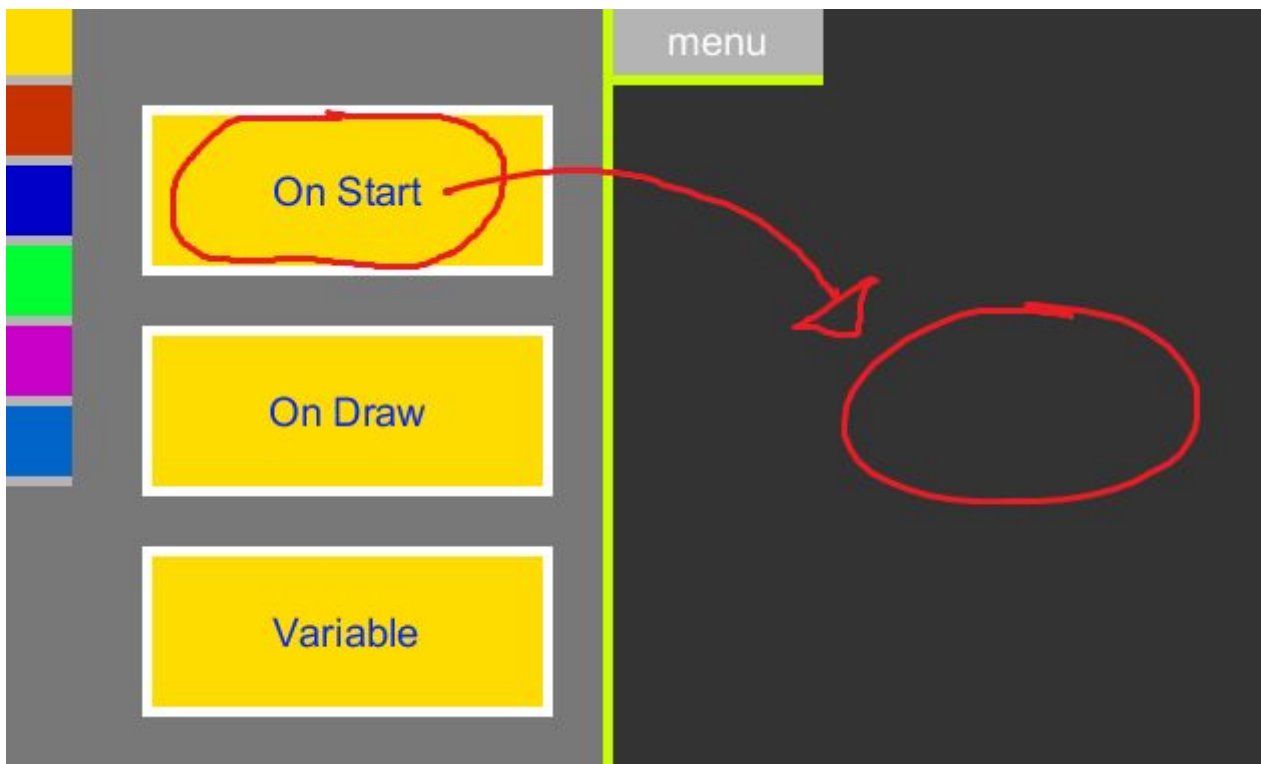
ProgBlock kann durch das Analysieren von Block-Klassenkomplexen ein Programm herleiten. Dieses Programm wird als Java-Processing-Datei in einem Ordner gespeichert. Es kann dann mit dem Processing-Editor geöffnet und als ein ausführbares Programm exportiert werden.

Einführen von neuen Funktionen

Alle Funktionen von den jeweiligen Blöcken sind in Tabellen-Dokumenten gespeichert und könnten von jemandem, der sich mit Java auskennt verändert oder erweitert werden. Das heißt, dass man theoretisch gesehen alles mögliche zu dem Programm hinzufügen könnte.

Aufbauen von Programmen

Um Programme mit ProgBlock zu schreiben muss man mit einem sogenannten “Eventblock” (die Farben sind diejenigen, die im Programm zu sehen sind) anfangen.

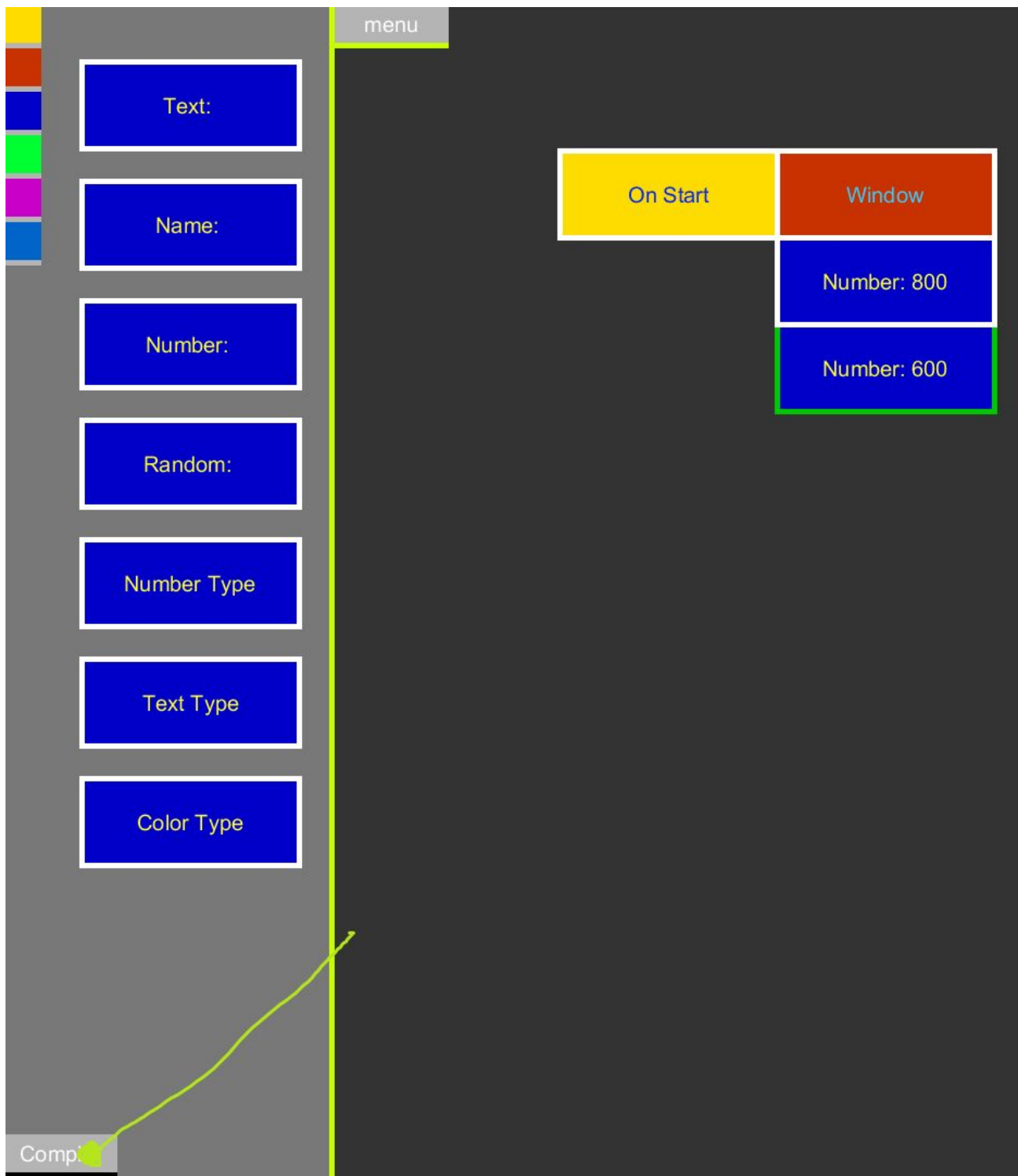


Daraufhin geht man zu den “Aktionen” und wählt beispielsweise Window, welches die Größe deines Programmfensters ändert und heftet dies an den “Eventblock”.

Für die Window Aktion muss man zwei Zahlen angeben. Dafür muss man zum Teil, genannt:



„Daten“, zwei Zahlenblocks nehmen und diese an „Window“ heften. Zum Ändern der Zahlen darin muss man es rechtsklicken.



Nach dem man die Zahlen so eingestellt hat, wie man sie haben möchte, sollte man auf das „Compile“ Feld unten links im Menü drücken und einen Namen eingeben.

Nun hat man ein Programm welches ein Fenster öffnet, welches die selbst eingestellte Größe hat.
Im Editor sieht das dann so aus:

```
test33
1 void setup() {
2   size(800, 600);
3 }
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

Done saving.

Einen neuen Block hinzufügen

Um einen neuen Block hinzuzufügen, muss man ein wenig Vorkenntnis in der Sprache Processing. es ist im Programmordner ein weiterer Ordner mit dem Namen "blocks". Darin sind Dateien, die alle Daten der Blöcke speichern.

Wenn man die benötigten Kenntnisse in Processing hat, kann man eine dieser Dateien öffnen und selbst einen Block hinzufügen. Aber wir haben eigentlich vor alles was mit der Sprache möglich ist, in unser Programm einzufügen.

Liste der Jetzigen Blöcke

Events

Events sind Blöcke, an die man andere anhängt. Zu Beginn des Programms hat man immer einen.

On Start: Dieser Block ist zu Beginn des Programms. Beim Beginn des Programms werden alle daran hängende Blöcke von links nach rechts ausgeführt.

On Draw: Alle an diesem Event angekoppelte Blöcke, werden von links nach rechts wiederholt ausgeführt.

Variable: Dieser Block erstellt eine Variable (Ein Namen mit bestimmten Wert). Zuerst gibt man einen Typ (s. Data) an, dann den Wert.

Actions

Actions sind Blöcke die generelle Befehle ausführen, wie die Fenstergröße deines Programms zu ändern.

Print: Schreibt alle angehängenen Textblöcke in die Konsole bei Processing rein.

Background: Macht den Hintergrund des Fensters die Farbe die angehängt ist.

Exit: Beendet das Programm

Fullscreen: Macht das Fenster Vollbild.

Window: Macht das Fenster so breit, wie die erste Zahl und so hoch wie die nächste.

Delay: Wartet eine Anzahl von Millisekunden.

Data

Data-Blöcke sind Blöcke die benutzt werden um Daten wie Zahlen oder Text zu speichern oder zu nutzen.

Text: Rechtsklick um Text einzugeben.

Name: Rechtsklick um einen Namen einzugeben.

Number: Rechtsklick um eine Zahl einzugeben.

Random: Rechtsklick um eine Zahl einzugeben. Dann wird eine zufällige Zahl zwischen dieser und null.

Number/ Color/ Text Type: Diese werden benutzt um Variablen einen bestimmten Typ zuzuordnen.

Drawing

Drawing Blöcke benutzt man um Formen und Bilder auf den Bildschirm zu malen.

Rectangle: Man gibt hier vier Werte an: die X Position, die Y Position, eine Breite und eine Höhe. Mit dieser Information macht es dann ein Rechteck wo die Koordinate die oben-linke Ecke ist.

Ellipse: Man gibt hier vier Werte an: die X Position, die Y Position, eine Breite und eine Höhe. Mit dieser Information macht es dann eine Ellipse wo die Koordinate das Zentrum ist.

Triangle: Man gibt hier sechs Werte an: drei X Positionen und drei Y Positionen abwechselnd. Mit diesen Positionen macht es dann ein Dreieck.

Quad: Man gibt hier acht Werte an: vier X Positionen und vier Y Positionen abwechselnd. Mit

diesen Positionen macht es dann ein Viereck.

Circle: Man gibt hier vier Werte an: die X Position, die Y Position, einen Radius. Mit dieser Information macht es dann ein Kreis wo die Koordinate das Zentrum ist.

Line: Man gibt hier zwei Werte an: zwei X Positionen und zwei Y Positionen abwechselnd. Mit diesen Positionen macht es dann eine Linie.

Point: Man gibt hier zwei Werte an: eine X Position und eine Y Position abwechselnd. Mit diesen Positionen macht es dann einen Punkt.

Operations

Operation Blöcke benutzt man für jegliche verschiedene Logik und für Mathematische Berechnungen.

Add: Man gibt eine Variable und eine Zahl an und diese wird dann zum Wert der Variable addiert.

Subtract: Man gibt eine Variable und eine Zahl an und diese wird dann vom Wert der Variable subtrahiert.

Multiply: Man gibt eine Variable und eine Zahl an und der Wert der Variable wird mit dieser multipliziert.

Divide: Man gibt eine Variable und eine Zahl an und der Wert der Variable wird durch diese geteilt.

Set: man gibt eine Variable und einen Wert an. Die Variable wird dann gleich dem Wert gesetzt.

Set Color: Man gibt eine Variable und drei Werte an. Die Variable wird dann gleich einer Farbe mit diesen Rot, Grün und Blau Werten gesetzt.

Equals: Wird in If Schleifen verwendet. Ist Wahr, wenn die Zwei angegebene Werte gleich sind sind.

Greater than: Wird in If Schleifen verwendet. Ist Wahr, wenn der erste Wert größer ist als der Zweite.

Smaller than: Wird in If Schleifen verwendet. Ist Wahr, wenn der erste Wert kleiner ist als der Zweite.

Control

Control Blöcke werden dazu verwendet Schleifen zu machen und Werte zu Testen.

If: Eröffnet eine Schleife und alles was Drinnen ist wird nur ausgeführt wenn die Aussage Wahr ist.

Repeat: Eröffnet eine Schleife und alles was Drinnen ist wird so oft ausgeführt wie die Angegebene Zahl.

Close Loop: Schließt die Jetzige Schleife

Pläne des Projekts

1. Wir wollen das Programm einfacher zu bedienen lassen.
2. Bessere Grafiken.
3. Mehrere Sprachen hinzufügen.
4. Mehr Basis Blöcke (die man nicht selbst hinzufügen muss)

Fazit

Unser Programm ist insgesamt ganz OK. Es hat Möglichkeiten zur Verbesserung und wäre eventuell auch für kleinere Projekte brauchbar, ist jedoch nicht gut genug für größere Produkte.