

Dokumentation zur Aufgabe 1 – Pizzavision



Programmdokumentation von
Christian Siewert
für den 27. Bundeswettbewerb Informatik

1 Aufgabenstellung

Die erste Aufgabe des BWINF besteht darin ein Bestellsystem für die Online-Bestellung einer Pizza zu realisieren. Dabei soll sich ein User aus genau 12 verschiedenen Zutaten seine Pizza zusammenstellen können. Außerdem soll ein Vorschaubild der fertigen Pizza existieren, welches laufend durch Useraktionen (Zutat hinzufügen / entfernen) aktualisiert werden soll. Es ist des weiteren darauf zu achten das die Pizzabilder optisch ansprechend aussehen, eine Vorfreude darstellen und nebenbei Appetit auf mehr Zutaten machen.

2 Lösungsidee

Es gibt nur wenig Teilprobleme über die man sich bei der Bearbeitung der Aufgabe Gedanken machen muss. So müssen natürlich erstmal geeignete Zutaten gefunden werden, welche später auf die Pizza gelegt werden können. Darüber hinaus sollten gut aussehende Bilder dieser Zutaten recherchiert (Internet oder selber fotografieren) werden. Mit diesen kann man dann später arbeiten um die Pizza zu belegen.

Der nächste Schritt wäre sich zu überlegen wie man die Bilder nun am besten in das Programm implementiert und dem User verschiedene Auswahlmöglichkeiten zur Verfügung stellt. Ich werde dies höchstwahrscheinlich mit Checkboxes realisieren. Wenn ein User eine solche Checkbox anklickt (z. B. die Checkbox für das Hinzufügen / Entfernen von Salamischeiben), dann soll ein Bild mit Salamischeiben angezeigt werden. Dabei werden die einzelnen Bilder (resp. Zutaten) übereinander gelegt und je nach Auswahl des Users angezeigt. Somit entsteht der Eindruck das der User ein Bild vor Augen hat, obwohl es viele verschiedene sind. Darüber hat der User immer vor Augen welche Zutaten auf seiner Pizza bereits drauf sind.

Die Bilder werden vor der Verwendung in einem Bildbearbeitungsprogramm (evtl. Gimp) bearbeitet. Dadurch wird eine schönere Optik erreicht und die Pizzen sollten in allen Belegungsstadien lecker aussehen.

3 Programm-Dokumentation

Ich habe mich für folgende 12 Zutaten entschieden:

- Tomatensoße
- Salami
- Gewürze
- Tomaten
- Schinken
- Pilze
- Zwiebeln
- Peperoni
- Thunfisch
- Ananas
- Paprika
- Oliven

Der Pizzaboden ist immer verfügbar und zählt in die Zutatenauswahl nicht hinein. Die Lösung wurde so umgesetzt wie in der Lösungsidee beschrieben. Es werden am Anfang boolsche (zutatenspezifische) Variablen (z. B. `pm_salami`) deklariert die mit dem Wert `false` initialisiert werden. Diese Variablen werden später benutzt um zu prüfen ob die jeweilige Zutat zu sehen ist oder nicht. Wie weiter oben beschrieben habe ich in meinem Programm 13 verschiedene Bildelemente implementiert auf denen später die Zutaten und der Boden angezeigt werden. Außerdem wurde zu jeder Zutat eine Checkbox eingebunden die ein User aktivieren oder deaktivieren kann. Die Bildelemente sind beim Starten der Applikation unsichtbar. Beim Aktivieren einer Checkbox wird das jeweilige Bildelement sichtbar und die Zutat wird entsprechend angezeigt. Die Bilder befinden sich auf der beigelegten DVD im Odner „.../Pizzavision/gui“ und enden alle mit dem Suffix „_final.png“.

Nachdem eine Zutat angezeigt wird (z. B. die Salamischeiben) wird die zugehörige boolsche Variable (`pm_salami`) auf `true` gesetzt. Wenn ein User nun diese Zutat wieder entfernen möchte, braucht er nur die jeweilige Checkbox zu deaktivieren. Dazu wird, wie beim Aktivieren auch, geprüft ob die jeweilige boolsche Variable auf `true` steht. Ist dies der Fall so wird das Bildelement auf unsichtbar gestellt.

4 Programmablauf-Protokoll

Im folgenden möchte ich einen Testlauf meiner Anwendung durchführen. Der Anfangszustand ist nach dem folgenden Bild definiert:



Wie man sieht ist nach dem Start der Anwendung keine Zutat ausgewählt und man sieht dementsprechend nur den Pizzaboden der immer angezeigt wird. Wenn man jetzt ein paar Zutaten hinzufügen will muss man nur die jeweilige Checkbox aktivieren. Danach wird, in Form einer Abfrage, geprüft ob die jeweilige Zutat schon zu sehen ist. Falls nicht wird diese angezeigt.



Wie man sieht habe ich die Checkboxen „Tomatensauce, Salami, Tomaten und Ananas“ aktiviert. Diese Zutaten werden dann entsprechend angezeigt. Im folgenden möchte ich die Ananas wieder entfernen. Ich deaktiviere dazu einfach die Checkbox für die Ananas:



Die Ananasscheiben wurden wie gewünscht entfernt. Der Benutzer liefert den Input also nur über das Aktivieren und Deaktivieren von Checkboxen.

5 Quellcode

```
#include <gtk/gtk.h>                // Den Gtk-Header implementieren. Ohne ihn läuft nichts ;-)
```



```
#define GUI "gui/pm.xml"           // Pfad zur GUI
```



```
typedef struct{                    // Die Hauptstruktur für das Programm --> besteht aus...
```



```
    GtkWidget      *win_main;      // ... Hauptfenster
```



```
    GtkWidget      *img_sauce;     // ... Bild für die Soße
```

```
    GtkWidget      *img_salami;    // ... Bild für die Salami
```

```
    GtkWidget      *img_tomaten;   // ... Bild für die Tomaten usw.
```

```
    GtkWidget      *img_ananas;
```

```
    GtkWidget      *img_pilze;
```

```
    GtkWidget      *img_peperoni;
```

```
    GtkWidget      *img_zwiebeln;
```

```
    GtkWidget      *img_paprika;
```

```
    GtkWidget      *img_fisch;
```

```
    GtkWidget      *img_gewuerze;
```

```
    GtkWidget      *img_schinken;
```

```
    GtkWidget      *img_oliven;
```



```
} PizzaMaker;                    // Name für die Struktur
```



```
gboolean pm_sauce = FALSE;        // Soße sichtbar?
```

```
gboolean pm_salami = FALSE;       // Salami sichtbar?
```

```
gboolean pm_tomaten = FALSE;      // Tomaten sichtbar? usw.
```

```
gboolean pm_ananas = FALSE;
```

```
gboolean pm_pilze = FALSE;
```

```
gboolean pm_peperoni = FALSE;
```

```
gboolean pm_zwiebeln = FALSE;
```

```
gboolean pm_paprika = FALSE;
```

```
gboolean pm_fisch = FALSE;
```

```
gboolean pm_gewuerze = FALSE;
```

```
gboolean pm_schinken = FALSE;
```

```
gboolean pm_oliven = FALSE;
```



```
// ----- Funktionsprototypen -----
```



```
GdkColor color = {0, 0xEEFF, 0xEBFF, 0xE6FF}; // Hintergrundfarbe für GtkWindow
```



```
int main (int argc, char *argv[]);          // Hauptfunktion
```



```
void on_win_main_destroy (GtkObject *object, PizzaMaker *program); // Funktion für das Beenden des Programms
```

```
void on_cke_sauce_toggled(GtkObject *object, PizzaMaker *program); // Klick auf CheckBox für die Soße
```

```
void on_cke_salami_toggled(GtkObject *object, PizzaMaker *program); // Klick auf CheckBox für die Salami
```

```
void on_cke_tomaten_toggled(GtkObject *object, PizzaMaker *program); // Klick auf CheckBox für die Tomaten usw.
```

```

void on_cke_ananas_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_pilze_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_peperoni_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_zwiebeln_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_paprika_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_thunfisch_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_gewuerze_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_schinken_toggled(GtkObject *object, PizzaMaker *program);
void on_cke_oliven_toggled(GtkObject *object, PizzaMaker *program);

gboolean build_app (PizzaMaker *program);    // Funktion zum Erstellen der Applikation

// ----- Funktionen -----

gboolean build_app (PizzaMaker *program){    // Funktion zum Erstellen und Anzeigen der Applikation

    GtkBuilder *builder;
        builder = gtk_builder_new ();
    gtk_builder_add_from_file (builder, GUI, NULL);

    program->win_main = GTK_WIDGET (gtk_builder_get_object (builder, "win_main"));
    program->img_sauce = GTK_WIDGET (gtk_builder_get_object (builder, "img_sauce"));
    program->img_salami = GTK_WIDGET (gtk_builder_get_object (builder, "img_salami"));
    program->img_tomaten = GTK_WIDGET (gtk_builder_get_object (builder, "img_tomaten"));
    program->img_ananas = GTK_WIDGET (gtk_builder_get_object (builder, "img_ananas"));
    program->img_pilze = GTK_WIDGET (gtk_builder_get_object (builder, "img_pilze"));
    program->img_peperoni = GTK_WIDGET (gtk_builder_get_object (builder, "img_peperoni"));
    program->img_zwiebeln = GTK_WIDGET (gtk_builder_get_object (builder, "img_zwiebeln"));
    program->img_paprika = GTK_WIDGET (gtk_builder_get_object (builder, "img_paprika"));
    program->img_fisch = GTK_WIDGET (gtk_builder_get_object (builder, "img_fisch"));
    program->img_gewuerze = GTK_WIDGET (gtk_builder_get_object (builder, "img_gewuerze"));
    program->img_schinken = GTK_WIDGET (gtk_builder_get_object (builder, "img_schinken"));
    program->img_oliven = GTK_WIDGET (gtk_builder_get_object (builder, "img_oliven"));

    gtk_widget_modify_bg(program->win_main, GTK_STATE_NORMAL, &color);    // Hintergrundfarbe setzen

    gtk_builder_connect_signals (builder, program);
        g_object_unref (G_OBJECT (builder));

    return TRUE;
}

int main (int argc, char *argv[]) {    // *** Hauptfunktion

    PizzaMaker *program;    // Ableitung der Struktur erstellen
    program = g_slice_new (PizzaMaker);    // Speicher für die Struktur anfordern

    gtk_init (&argc, &argv);    // GTK+ initialisieren

```

```

if (build_app (program) == FALSE) return 1; // War das Erstellen der Applikation erfolgreich?

gtk_widget_show (program->win_main);          // Das Hauptfenster (win_main) anzeigen

gtk_main ();                                  // Den Gtk-Event-Loop ausführen

g_slice_free (PizzaMaker, program);           // Die Speicherbereiche die belegt waren wieder freigeben

return 0;
}

void on_win_main_destroy (GtkObject *object, PizzaMaker *program){          // *** Funktion für das Beenden des Programms

    gtk_main_quit();                // Hier wird die Funktion zum Beenden der Applikation aufgerufen
}

void on_ckb_sauce_toggled(GtkObject *object, PizzaMaker *program){          // Klick auf CheckBox für die Soße

    if(pm_sauce == FALSE) {    gtk_widget_show (program->img_sauce);        pm_sauce = TRUE; }          // Wenn Bild
    nicht sichtbar dann zeige Bild an
    else                        {      gtk_widget_hide (program->img_sauce);    pm_sauce = FALSE; }          //
    ansonsten mache es unsichtbar
}

void on_ckb_salami_toggled(GtkObject *object, PizzaMaker *program){          // Klick auf CheckBox für die Salami

    if(pm_salami == FALSE) {    gtk_widget_show (program->img_salami);        pm_salami = TRUE; }
    else                        {      gtk_widget_hide (program->img_salami);    pm_salami = FALSE; }

}

void on_ckb_tomaten_toggled(GtkObject *object, PizzaMaker *program){          // Klick auf CheckBox für die Tomaten usw.

    if(pm_tomaten == FALSE) {    gtk_widget_show (program->img_tomaten); pm_tomaten = TRUE; }
    else                        {      gtk_widget_hide (program->img_tomaten); pm_tomaten = FALSE; }

}

void on_ckb_ananas_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_ananas == FALSE) {    gtk_widget_show (program->img_ananas); pm_ananas = TRUE; }
    else                        {      gtk_widget_hide (program->img_ananas); pm_ananas = FALSE; }

}

void on_ckb_pilze_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_pilze == FALSE) {    gtk_widget_show (program->img_pilze); pm_pilze = TRUE; }
    else                        {      gtk_widget_hide (program->img_pilze); pm_pilze = FALSE; }
}

```



```

}

void on_cke_peperoni_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_peperoni == FALSE) { gtk_widget_show (program->img_peperoni); pm_peperoni = TRUE; }
    else                        { gtk_widget_hide (program->img_peperoni); pm_peperoni = FALSE; }

}

void on_cke_zwiebeln_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_zwiebeln == FALSE) { gtk_widget_show (program->img_zwiebeln); pm_zwiebeln = TRUE; }
    else                      { gtk_widget_hide (program->img_zwiebeln); pm_zwiebeln = FALSE; }

}

void on_cke_paprika_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_paprika == FALSE) { gtk_widget_show (program->img_paprika); pm_paprika = TRUE; }
    else                     { gtk_widget_hide (program->img_paprika); pm_paprika = FALSE; }

}

void on_cke_thunfisch_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_fisch == FALSE) { gtk_widget_show (program->img_fisch); pm_fisch = TRUE; }
    else                   { gtk_widget_hide (program->img_fisch); pm_fisch = FALSE; }

}

void on_cke_gewuerze_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_gewuerze == FALSE) { gtk_widget_show (program->img_gewuerze); pm_gewuerze = TRUE; }
    else                     { gtk_widget_hide (program->img_gewuerze); pm_gewuerze = FALSE; }

}

void on_cke_schinken_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_schinken == FALSE) { gtk_widget_show (program->img_schinken); pm_schinken = TRUE; }
    else                      { gtk_widget_hide (program->img_schinken); pm_schinken = FALSE; }

}

void on_cke_oliven_toggled(GtkObject *object, PizzaMaker *program){

    if(pm_oliven == FALSE) { gtk_widget_show (program->img_oliven); pm_oliven = TRUE; }
    else                    { gtk_widget_hide (program->img_oliven); pm_oliven = FALSE; }

}
}

```