

TikzEdt Doku und TODO-Liste

1 MainWindow und UI

Aufgaben der Komponente:

- Integriert alle anderen Teile des Programms.
- Sieht gut aus.

ToDo:

- sehr vieles,... muss aber z.T. noch warten, bis die anderen Komponenten funktionieren.
- Gutes Icon fuer TikzEdt erfinden.

1.1 Der Code-Editor auf der Hauptseite (avalonedit)

ToDo:

- Syntax-Highlighting konfigurieren. Konkret: xml-Datei TikzSyntax.xshd vervollstaendigen.
- (Evtl.) Auto-Vervollstaendigung implementieren. (Komplizierter, be-noetigt je nach Perfektionsanspruch den Parser)

2 Klasse TikzDisplay (Kompilierung und Anzeige des Bildes)

Aufgaben der Komponente:

- Nimmt einen string "code", kompiliert den und zeigt ihn an. (Methode Compile)
- Erzeugt events (event OnCompileEvent) um den Status anzuzeigen.

Die Kompilierung und Anzeige funktioniert groessenteils. Details muessen noch verbessert werden.

ToDo:

- Fehlermeldungen von pdflatex abfangen und mit ausgeben. (Am besten mit Zeilennummer, Spaltennummer)
- Manchmal blockiert pdflatex bei einem Fehler anstatt mit Fehlermeldung abzuberechnen. Loesen oder im schlimmsten Fall den Prozess nach Timeout killen. (Anmerkung: das gleiche Phaenomen gibt es bei ktikz.)

3 Klasse TikzParser (Parsen von TikzCode)

Aufgaben der Komponente:

- Hat nur eine relevante Methode:

```
Parse(string code)
```

Parsed den code und erzeugt einen Parsetree (Klasse

```
Tikz_ParseTree
```

) oder null im Fehlerfall. (Anmerkung: Ist eine "best effort" Komponente. Es wird nicht jede Tikz-Konstruktion erkannt.)

Ich habe eine Antlr-Grammatik geschrieben, die einfache Tikz-Files parsen kann.

ToDo:

- Man muss den parse-tree den antlr liefert noch zu einem Tikz_ParseTree verarbeiten.
- Man kann beliebig viel Zeit investieren, Tikz noch genauer abzubilden.

3.1 Klasse Tikz_ParseTree

Aufgaben der Komponente:

- Enthaelte den ParseTree.

ToDo:

- Muss groesstenteils noch geschrieben werden. Es steht nur ein Geruest.
- Insbesondere fehlt eine Methode GetBB() die ein estimate der Bounding-Box liefert. (=maximum der vorkommenden Koordinaten)

4 Klasse PdfOverlay

Aufgaben der Komponente:

- Halbtransparentes Control, das ueber dem Pdf-output angezeigt wird.
- Nimmt einen

`Tikz_ParseTree`

und zeigt die enthaltenen Knoten (und Kanten?) an.

- Enthaelte Editierfunktionen: (i) Dinge auswaehlen und verschieben (ii) Nodes hinzufuegen (iii) Kanten hinzufuegen (iv) +X
- Bei Aenderung des ParseTrees wird ein Event ausgeloeet. (Die Mainwindow muss daraufhin den code im editor aktualisieren)

Bisher zeigt die Klasse ein paar Knoten an, die man verschieben kann, aber es wird noch kein Code editiert. (Ist also ein Mockup)

ToDo:

- Muss groesstenteils noch geschrieben werden.

5 SnippetManager????

Man sollte dem vergesslichen Nutzer (wie mir) noch helfen, sich an Befehle etc zu erinnern. Ich weiss noch nicht, wie genau man das anstellt. Ein Vorschlag: Ein Snippetmanager. Enthalt eine Bibliothek von Snippets, die per Menu oder Toolbar oder PopupFenster etc. abgerufen werden koennen und an die Cursorposition eingefuegt werden.

Einige Ideen (Brainstorming)

- Jedes Snippet enthseelt (i) den SnippetText (klar) (ii) es hat eine Kategorie, (z.B. Template fuer ganze Vorlagen, Edge/Vertex-Style fuer Stye-Befehle, etc...) und (iii) ein einfaches komplettes tikzpicture, das die Benutzung einfachstmoeiglich demonstriert. Das Picture kann man rendern und neben dem Snippet als Bild anzeigen.
- Der Benutzer kann Snippets hinzufuegen. Das ersetzt insbesondere den "Stylemanager" aus GraphToTikz, und ausserdem eine Vorlegenverwaltung.

ToDo:

- Muss noch entworfen und geschrieben werden.
- Was ist die beste Realisierung? Ist es noetig?