Generelle Informationen

- Englische Kommentare Stammen von mir
- Ich hab sehr viel Zeit zuhause verbracht die Sprache zu lernen und verstehen

Das Programm bietet folgende Funktionen:

- Speichern und Öffnen der Möbelobjekte
- Selektion von Möbelobjekten durch einfaches klicken
- Bewegen der Möbelstücke durch Drag and Drop
- Durch scrollen können Möbelobjekte rotiert werden
- Bei betätigter STRG-Taste beim scrollen kann die Gräößе verändert werden
- Alle manuell veränderten Attribute können auf den Standardwwer zurückgesetzt werden
- Menüleiste / Menubar mit Icons
- Das Ändern der Farbe eines Möbelonjekts
- Und vieles mehr...

(In den Logbucheinträgen sind nicht alle Details festgehalten; die meisten Erklärungen befinden sich im Code selbst)

Logbucheinträge

Donnerstag 09/01/2020 Informatikunterricht

Änderungen: - Farbänderungen bei Auswahl von einem Objekt zu schwarz

Heute habe ich versucht die Funktion des Fabwechsels mit einhergehender Auswahl fertigzustellen. Diese sind noch nicht voll funktionsfähig. Leider bin ich an einen Kopellierfehler hängen geblieben. Ich habe noch nicht ganz verstanden, was den Fehler auslöst. Deshalb bin ich leider etwas frustriert.

1Donnerstag 6/01/2020 Informatikunterricht

Den Fehler aus der letzen Unterricht konnte ich beheben: Der Fehler lag in der Änderung der Farbe, welche ohne weiteres nicht direkt geändert werden konnte. Deshalb hab ich eine Methode zum Ändern der Farbe in der Furniture Klasse erstellt. Ich habe eine neue Variable erstellt, welche den Farbzustand des Objekts speichert, sodass ich bei der Deselektion des Objekts den vorherigen Farbzustand wiederherstellen kann.

Den Rest der Unterrichtszeit habe ich damit verbracht mir anzuschauen wie die JMenuBar funktioniert und die Bar in meinem Programm implementiert. Dabei habe ich noch nicht ganz verstanden, wie die dahinterstehende Funktion ausgelöst wird. In den Beispielen wird nämlich eine kompliziert aussehende eigene Klasse erstellt, welche mit einem ActionListener erweitert wird. Die JMenuItems führen also noch nichts aus und sind somit "dumm".

Dienstag 21/01/2020

Ich habe heute die grundlegenden Funktionen der GUI.java in Leinwand.java implementiert (sprich main function und JPanel). Da die Funktionen von dem vorherigen JButton Panel (u.a. Objekte erzeugen und Objekte bewegen) nicht mehr benötigt werden und diese Funktionalität durch JMenuBAr übernommen wurde. Bei dem Prozess musste ich einige Methoden anpassen oder teilweise auch neue, bessere oder effizientere Methoden schreiben. Bei der Übernahme hatte ich teilweise Probleme mit statischen und nicht-statischen Funktionen und da häufig eine Kette von Methoden voneinander abhängig ist wirkt sich eine Veränderung (z.B. Veränderung von statisch zu nicht-statisch) sofort auf alle anderen aus. Dies macht die Fehlersuche schwierig.

Datum unbekannt: Ab hier habe ich mich regelmäßig nachmittags hingesetzt und weitergearbeitetet

Ich habe heute ziemlich viel Zeit mit der Drag and Drop Funktion verrbracht. Die Schwierigkeit liegt in dem sofortigen verschieben des Objects, da mit einer Bewegung der Maus auch eine Bewegung des dementsprechenden Objekts auslösen soll. Ich habe mich an einem Codeblock von Stackoverflow orientiert, welchen ich auch als Demo in meinem Programm eingebaut habe. Jedoch musste ich noch eine Menge Modifikationen durchführen, um diesen mit meinen Raumplaner funktioniert, da der Codeblock davon ausgeht, dass das verschobene Objekt nach dem verschieben an einer neuen Stelle ist. Beim Raumplaner kann man allerdings den ActionListener nur zur Leinwand hinzufügen und da sich dieses Objekt nicht bewegt musste ich eine Veränderung duchführen. Beim Klicken wird die Position des Möbelstücks sowie die Position des Cursors in einer Variable festgehalten. Sobald man die Maus bewegt wird die neue Position des ausgewählten Objekts wie folgt festgelegt: Position des Objekts beim Klickvorgang + Aktuelle Cursorposition - Cursorposition beim Klickvorgang. Somit wird also im Prinzip die Differenz der von der Maus zurückgelegten Strecke zu der Position addiert/subtrahiert. Und dies passiert bei jedem Mal, wenn der MouseListener eine Bewegung registriert - so kommt das Drag and Drop Gefühl zu Stande.

Neuer Tag (Datum unbekannt)

Ich habe heute den ColorChanger implementiert, dabei habe ich ein kurzes YouTube Tutorial geschaut und danach verstanden wie dies funktioniert. Um die Farbe auch abspeichern zu können, habe ich den Typ des color Attributs von Typ String auf Typ Color geändert. Damit man wieder auf die ursprüngliche Farbe zurückkehren kann, speichert jedes Objekt zudem noch die Farbe beim erstellen des Objekts.

Zudem habe ich heute den MouseWheelListener so implementiert, sodass bei Bewegung des Mausrads das Objekt gedreht wird. Die Idee stammt von dem Raumplaner, welcher im Informatik Ordner als Vorbild abgelegt war. Der MouseWheelListener gibt entweder 1 oder -1 aus, abhängig von der Richtung. Diesen Wert kann ich einfach mit 10 multiplizieren und diesen dann zum aktuellen Rotationswert addiereDamit auch die Größe geändert werden kann habe ich die einfache Option hinzugeüft, dass beim gedrückt halten des STRG-Knopfes die Größe entweder vergrößert oder verkleinert wird. Dabei wird der aktuelle Größenwert des Objekts durch 10 geteilt und dann entweder addiert oder subtrahiert. Das bedeutet also, dass entweder 10 addiert oder subtrahiert wird.

Neuer Tag (Datum unbekannt)

Heute habe ich die Speicherfunktion implementiert. Zur Hilfe habe ich einige Anleitungen sowie Beispiele mir angeschaut. Dabei habe ich auch einen Codeblock übernommen und stark modifiziert, sodass er passt. Vor

dem Speichern wird überprüft, ob es eine derartige Datei bereits gibt und es wird eine Dateiendung angehängt, sofern dies nicht bereits von Nutzer passiert ist. Zudem müssen hier einige try/catch Blöcke verwendet werden, damit bei einem Fehler das Programm nicht abstürzt, sondern nur die jeweilige Funktion mit einem Fehler beendet. Beim Speichern werden einfach alle Objekte mit einem for-loop abgespeichert. Das Speichern funktionierte am Ende auch taddellos, allerdings wird beim laden nur das erste Objekt geladen.

Freitag 21/02/2020

Ich konnte heute den Fehler des Abspeicherns lösen: Der EOFFile Boolean löste zu früh aus und somit wurde immmer nur das erste Objekt importiert. Durch Veränderung des Mechanismus zur Erkennung, ob die Datei zu Ende ist konnte ich das Problem lösen. Heute habe ich das Aussehen etwas durch das hinzufügen von Icons aufpoliert. Damit die Icons nicht gigantisch angezeigt werden, musste ich mithilfe eines Codeblocks von Stackoverflow die Icons vorher auf die richtige Auflösung herunterskalieren. Die Icons verbessern das Erlebnis erheblich. Zudem habe ich noch ein About Reiter sowie die MouseMovementDemo in mein Programm integriert.

Verwendete Hilfestellung

Dabei habe ich mithilfe folgender (Video-) Tutorials gearbeitet und teilweise Code Blöcke daraus übernommen:

- Die offiziellen Java Dokumentation Java Docs
- Dieses Video half mit mit dem JMenu Youtube
- Ich habe den Codeblock vom MouseListener hier entnommen: Stackoverflow
- Hilfe mit dem JFileChooser und Codeblöcke Youtube
- Hilfe mit dem JFileChooser-Filtering Stackoverflow
- Codeblock für die infoBox Stackoverflow
- Zum speichern in einer Datei von Coderanch
- Grundlegende Erkennung der MouseRadRotation abgeschaut von Stackoverflow
- Filefilter, welcher die Endung anhängt: Stackoverflow
- Hilfe mit der JMenuBar von Stackoverflow
- Dieser Codeblock transformiert die Icons auf die richtige Größe Stackoverflow
- Die Icons stammen von Icons8