Jabberpoint wijzigingen

Rick Vinke – 5035678

# Inleiding

Voor OGO moest ik een applicatie genaamd Jabberpoint refactoren.  
Dit betekend de code opruimen en verbeteren zonder dat dit functionaliteit verandert.

In dit korte verslag vertel ik over wat ik ben tegen gekomen in de code en hoe ik wijzigingen heb aangebracht.

# Packages

Ten eerste heb ik de code ondergebracht in packages. Hierdoor kon ik de code later in het proces beter verdelen en werd het overzichtelijker. Dit maakte wel eerst het uitvoeren van het jar bestand buiten de IDE kapot. Maar dit was snel opgelost door het MANIFEST naar de goede class en package te wijzen.

# Opslaan en laden

Al gelijk ging ik kijken naar de code van het opslaan en laden van bestanden. Hier zag ik er een FileAccessor class was die beide het opslaan en laden regelde. De Demo en XML presentaties werden geladen door classen die de FileAccessor extende. Ik was toen al snel van plan om dit te splitsen en aparte saver en loader classen.

Ik heb toen een nieuwe files package gemaakt en daarin weer loading en saving packages. Hierin deed ik dan abstracte FileLoader en FileSaver klassen. Dit deed ik omdat zoals de vorige opstelleng, de XML en Demo classen dit moeten overnemen maar dat ze ook al functionaliteit moeten bevatten.  
Maar beide FileSaver en FileLoader kregen wel wat dezelfde functies zoals voor de extentie en sneltoets. Ik koos er toen voor om deze functies in een FileAccessor interface te doen, en de FileSaver en FileLoader abstrace klassen dit te laten implementeren.

Hierna heb ik de XMLAccessor en DemoPresentation herschreven om te werken met mijn nieuwe FileSaver en FileLoader systeem. Daarna schreef ik een FileHandler class die de savers en loaders dynamisch inlaad in sets waardoor ze aangeroepen kunnen worden. Als er in de toekomst ene nieuwe loader of saver moet worden toegevoegd, dan kan dit makkelijk vanuit hier.

Ook heb ik de MenuController aangepast om de savers en loaders dynamisch toe te voegen. Wanneer er maar 1 saver of loader is dan zie je de huidige functionaliteit. Maar als er meer dan 1 is dan wordt het Open of Save menu een submenu met de meerdere extentie opties.

# Styles

Toen ik de code door keek, toen zag ik ook dat de Slide styling hard-coded was. En dus niet kon worden aangepast. Dit wou ik dynamisch maken, zodat het makkelijk is als iemand in de toekomst ooit een Slide een ander lettertype of kleur wil geven.

Ik heb eerste de Style class herschreven. Dit was eerst heel static gefocust zodat je deze stijlen vanaf overal kon aanroepen. Maar dit zou niet werken als dit dynamisch zou worden en zou ook “bad practice” zijn. Ik heb een HashMap toegevoegd die het level (int) als key pakt en een nieuwe LevelStyle class als value. Hierdoor is er ook geen limitatie tot 5 item levels meer en kunnen er eventueel ook levels worden weggelaten als dit niet nodig is.

De nieuwe LevelStyle class lijkt heel erg op de oude Style class met de argumenten. Het is in principe een storage class die variables vast houd. De uitzondering is hoe het met het font omgaat omdat die het Font object genereerd van de gegeven argumenten. Ook neemt het de toString methode over van de oude Style class. Tot slot is er een DefaultStyle class toegevoegd die de Style class extend. Dit bevat de styles van de oude Style class en wordt op dit moment gebruikt voor alle slides.

Ik heb toen de Slide klassen aangepast om inplaats van de Style statisch aan te roepen, een Style argument te vereisten in de constructor. Hierdoor kan iedere Slide een aparte stijl hebben.\

# Overige kleinere veranderingen

Ik ben langs de code gegaan en heb onnodige variables en casts verwijdert en oudere code vervangen door betere code van nieuwere Java versies wanneer dit mogelijk was.

In de Slide klasse kwam ik tegen dat SlideItems in een Vector collectie werden opgeslagen. Hier had ik nog nooit eerder van gehoord dus dit heb ik opgezocht. Ik kwam er toen achter dat dit een verouderde collectie is die wordt afgeraden vanwege de trage snelheid. Daarom heb ik toen de Vector verandert naar een ArrayList en de getters en setters hierop aangepast.