

# 1 Problemen

## 1.1 Problemen

Een probleem dat we ondervonden was het drag-and-drop systeem maken. We hadden de mogelijkheid om drag-and drop functionaliteit van Java te gebruiken [?]. Echter is het doel van de drag-and drop van Java anders dan hetgeen de IDE moet hebben. De Java drag-and drop wordt gebruikt om componenten te verplaatsen tussen verschillende paneels, de IDE moet componenten verplaatsen en nesten binnen eenzelfde paneel.

Hierdoor hebben we ervoor gekozen om zelf drag-and-drop te implementeren. De implementatie details vind u in Sectie ???. De IDE berekent op basis van waar de gebruiker een blok sleept, of (en hoe) deze blok genest moet worden.

Een ander probleem was het invoegen van de implementatie van het opslaan en het compileren. Het opslaan (en het compileren) van blokken zijn losgekoppeld van de blokken zelf (de logica). Hoe dit gerealiseerd werd, was toch even een probleem waar we over moesten nadenken. Hiervoor hebben we gekozen om het visitor patroon toe te passen [?].

Axeski plaatst geen limitaties op het aantal processen dat tegelijkertijd kan runnen. Enkele duizenden processen tegelijkertijd runnen is geen probleem voor onze applicatie, echter indien de gebruiker een miljoen processen tracht te runnen zal de applicatie beginnen laggen, of indien de gebruiker nog meer processen runned, kan de applicatie vasthangen. Echter reageert onze applicatie net zoals een computer reageert. Deze kan ook vele processen tegelijkertijd runnen, maar teveel processen runnen zal de computer doen laggen.

Bij het uitprinten van grote hoeveelheden tekst naar de console bleef de GUI niet responsief. Dit werd opgelost door een aparte thread in te voeren die een buffer bevat waarin de tekst wordt opgeslaan. Deze buffer stuurt de text met een lagere frequentie door naar de console. Op deze manier blijft de GUI responsief.

## 1.2 Feedback verwerken

Er zijn verschillende aanpassingen gemaakt op basis van feedback die we gekregen hebben van zowel de begeleider als de opdrachtgever. Er zijn verschillende implementatiedetails die gebruiksvriendelijker geworden zijn door de feedback die we gekregen hebben.

Een voorbeeld hiervan is het mechanisme om blokken te verwijderen in het Klasse-view. Eerst stond er in het Klasse-view een vuilbak waarop de gebruiker zijn te-verwijderen blokken kon slepen. Dit werd een niet-passend mechanisme gevonden. Als reactie op deze feedback hebben we deze vuilbak verwijderd en de mogelijkheid geboden om blokken te verwijderen door een optie te selecteren

in het rechtermuisklik-menu.

Ook hebben we feedback verkregen over het kleurenschema dat we konden gebruiken. Door nieuwe combinaties te proberen hebben we iets gevonden wat neutraal en professioneel is.