



# FlexChess

- C++ Projekt mit OpenGL nutzung
- Texturierte Modelle mit Blender modelliert
- Variables Regelwerk (XML)

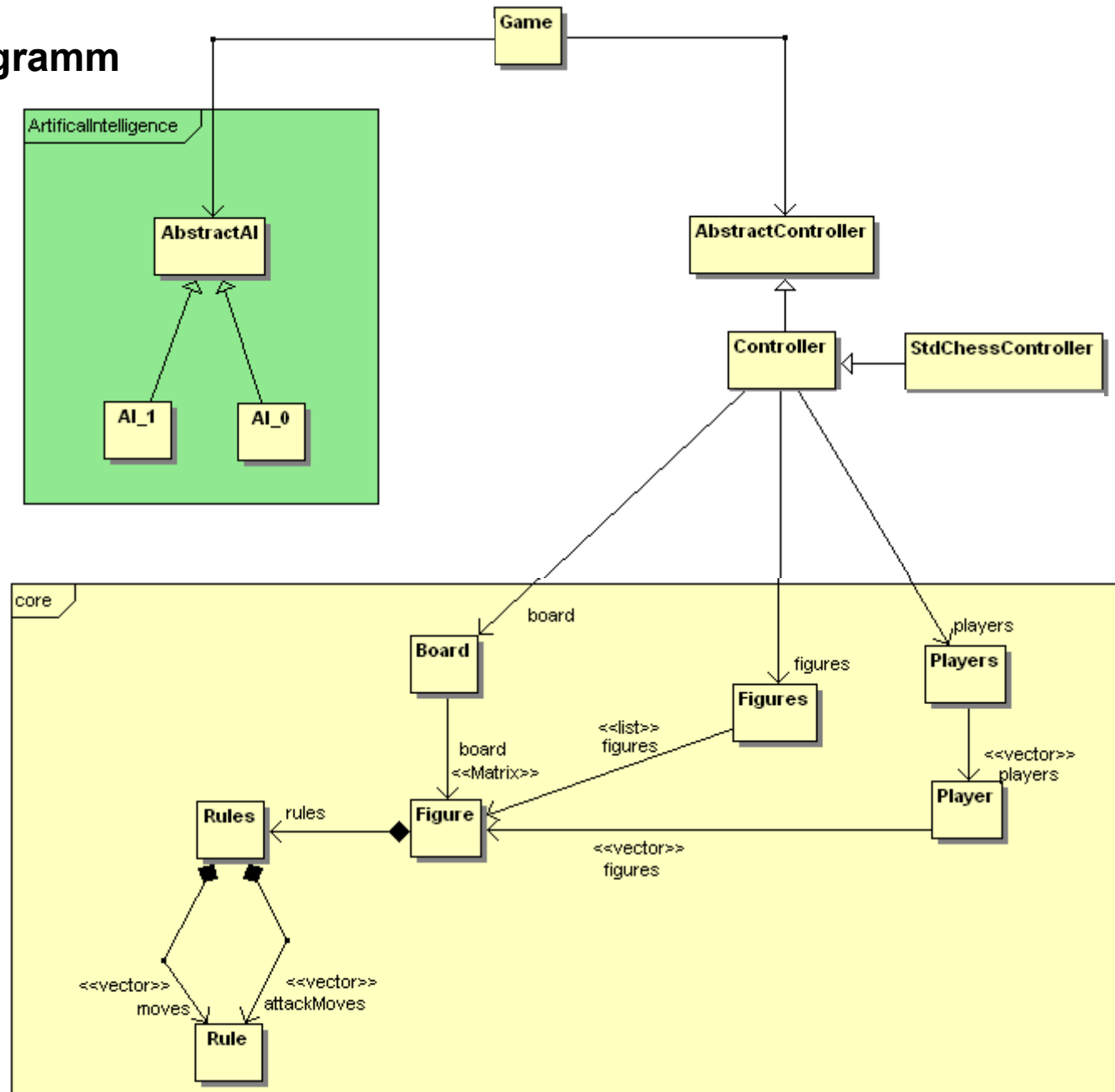
Veröffentlicht unter <https://github.com/sizilium/FlexChess>



## Es gibt zwei Möglichkeiten für Regel Änderungen:

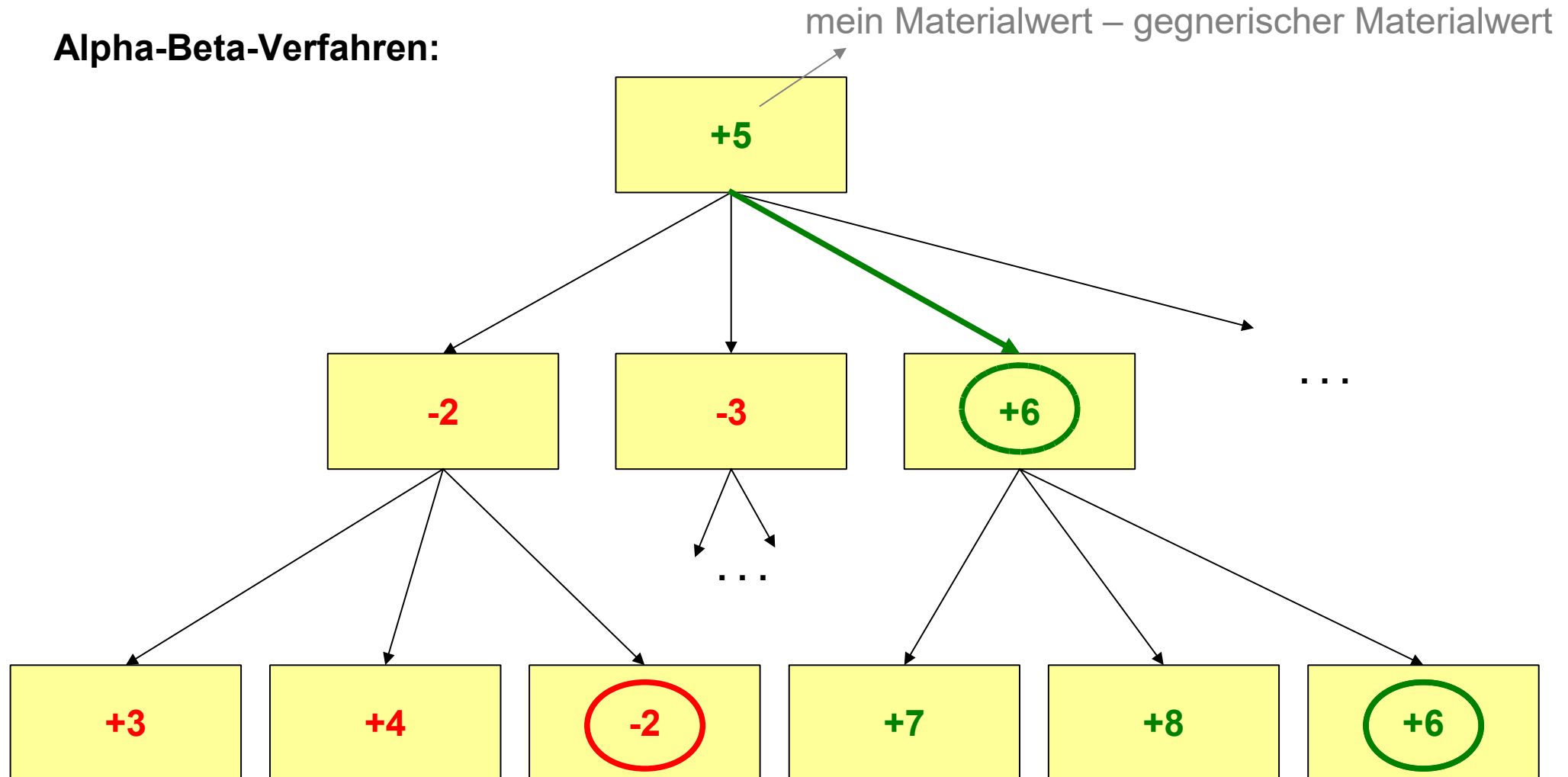
1. XML Dateien Editieren  erfordert nur ein Neustart des Programms
  - Startaufstellung (*formation.xml*)
  - Eigenschaften der Figuren (*figures.xml*)
  - Spielbrett Eigenschaften (*board.xml*)
2. Eigenen „Controller“ implementieren  das Programm muss neu kompiliert werden
  - von AbstractController Interface ableiten
  - eigene Regeln implementieren ...

## UML Klassendiagramm



## Implementiere Algorithmen:

- **AI\_0** ist ein Zufallsgenerator (debugging)
- **AI\_1** arbeitet nach dem Alpha-Beta-Algorithmus (Minimax Prinzip)
  - schaut zwei Halbzüge in die Zukunft
  - maximiert die eigene Punktzahl
  - minimiert die gegnerische Punktzahl
  - Punktzahl ist setzt sich aus Materialwert der Figuren zusammen
  - bei mehreren gleichen Ergebnissen entscheidet der Zufall

**Alpha-Beta-Verfahren:**

- die allgemeine KI weiter entwickeln
  - nicht nur Materialwert beachten sondern auch den Stellungswert der Figuren
  - auf eigene „Beweglichkeit“ achten
  - auf „Angriffsmöglichkeiten“ achten
  - das Alpha-Beta-Verfahren Rekursiv (beliebige Tiefe einstellbar) implementieren
- spezielle KI für die Standard Schach-Regeln implementieren
  - Stellung des eigenen Königs schützen
  - den gegnerischen König um jeden Preis Schachmatt setzen
  - Rochade einplanen, Fähigkeiten des Bauers einplanen
  - kleine Datenbank für die ersten Start-Züge einbauen
- das Spiel erweitern für bis zu vier Spieler auf einem Spielbrett!
- „schwebende“ Figuren implementieren („*hover*“ Eigenschaft in *figures.xml*)

**<https://github.com/sizilium/FlexChess>**