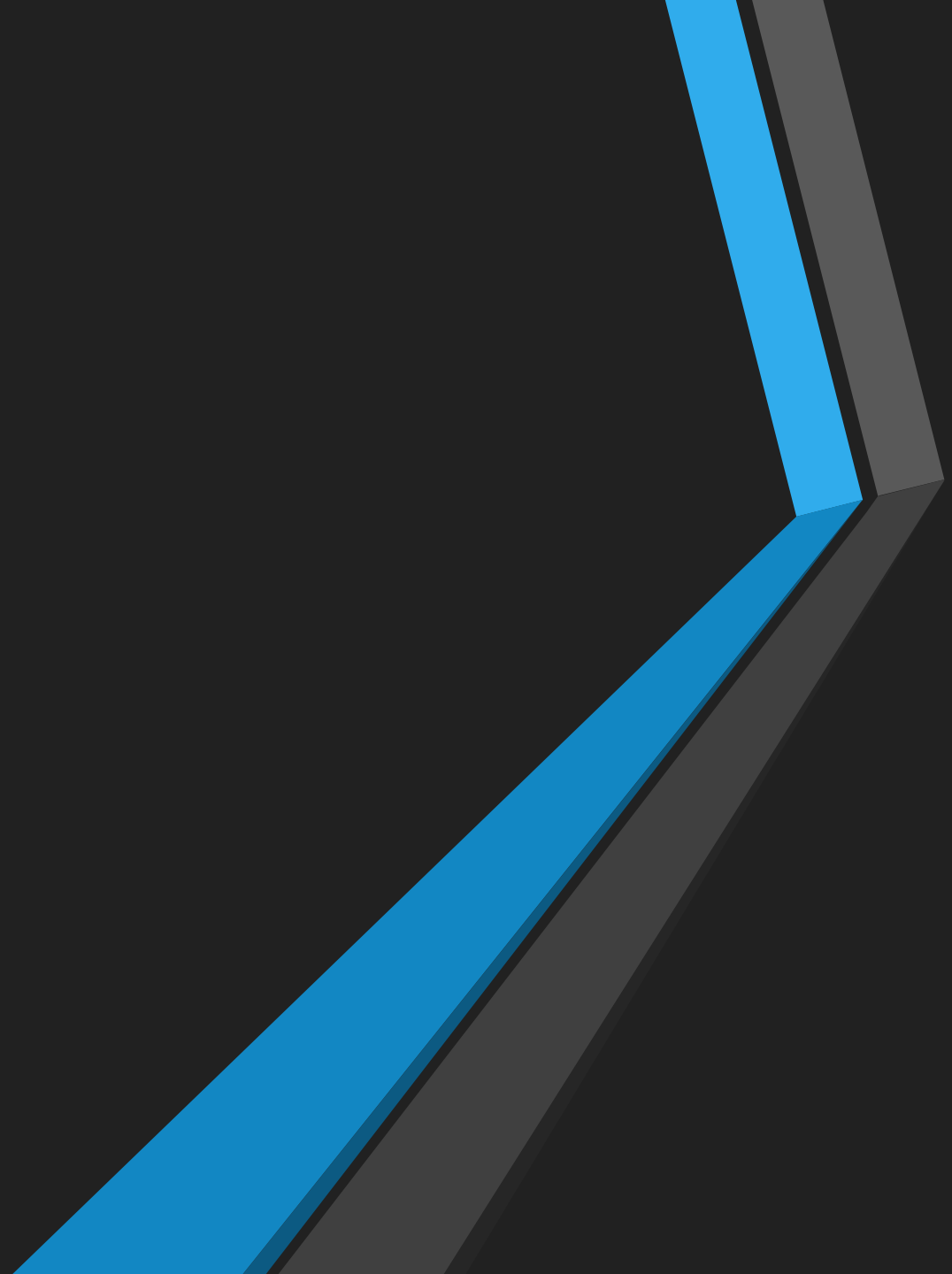


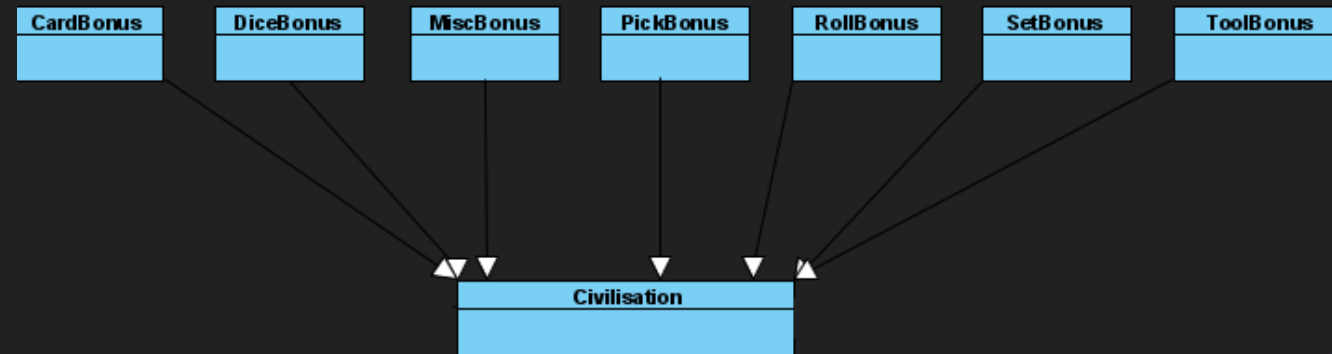
Stone Age

Bernd Uijtdebroeks & Denzell Mgbokwere



Polymorphism : civilisations

- Code duplicatie
- addItems



Low Coupling: Board

- Funties simple/geen parameters
- Geen dependencies

```
Board

-m_ended : bool
-m_currentPlayer : Colour
-m_buildingsCardStacks : std::vector<Building>[4]
-m_civilisationCards : std::vector<Civilisation>
-m_openCivilisationCards : std::vector<Civilisation>
-m_players : Player[4]
-m_hut : Hut
-m_forest : Gather
-m_clayPit : Gather
-m_quarry : Gather
-m_river : Gather
-m_hunt : Hunt
-m_toolShed : Toolshed
-m_field : Field
-m_round : int
-m_pickWindows : std::vector<PickRolled*>

+resetWorkers() : void
+feedWorkers() : void
+getOpenBuildingCard(pos : int) : Building*

+rerollBuildings() : void
+buildBuilding(colour : Colour) : void
+newOpenCivCards() : void
+addOpenCivCard() : void
+nextPlayer(cheked : int) : void
+payResources(colour : Colour) : void
+addRound() : void
+getToolShed() : ToolShed*
+getHut() : Hut*
+getField() : Field*
+getPlayer(colour : Colour) : Player*
+getGather(resource : Resource) : Gather*
+getRound() : int
+getCurrentPlayer() : Colour
+load(json : QJsonObject) : void
+save() : QJsonObject
+getEnded() : bool
+checkStacks() : bool
+end() : void
+checkChosen(colour : Colour) : void
+getOpenCivilisationCard(pos : int) : Civilisation*
+civilizeCivilisation(colour : Colour) : void
+newCivCards() : int
newBuild(building : Building*, pos : int) : void <<signal>>
allWorkersPlaced() : void <<signal>>
workersReset() : void <<signal>>
roundChanged() : void <<signal>>
newCiv() : void <<signal>>
-newBuilding(place : int) : void
```

Information Expert : Widgets

- Hebben nodige informatie

PayCiv
-m_player : Player*
-m_civ : Civilisation*
-m_amount : int
-m_hasPaid : bool
-ui : Ui::PayCiv*
+getHasPaid() : bool
-editText() : void <<slot>>
-on_acceptButton_clicked() : void <<slot>>
-on_cancelButton_clicked() : void <<slot>>
-totalSet() : int
-removeResources() : void

High cohesion: Player

- Functies over player
- Information expert

```
Player
-m_foodCount : int
-m_woodCount : int
-m_clayCount : int
-m_stoneCount : int
-m_goldCount : int
-m_buildingCount : int
-m_scoreCount : int
-m_foodGain : int
-m_colour : Colour
-m_freeWorkers : int
-m_workers : int
-m_tools : Tool[3]
-m_extraTools : std::vector<int>
-m_civBonuses : int[6]
-m_farmerCount : int
-m_toolMakerCount : int
-m_hutBuilderCount : int
-m_shamanCount : int

+addResource(resource : Resource, amount : int) : void
+addWorker() : void
+addTool() : void
+addScore(amount : int) : void
+addFoodGain() : void
+addBuilding() : void
+addExtraTool(tool : int) : void
+addBonus(farmers : int, makers : int, builders : int, shamen : int, bonus : CivBonus)
+getResource(resource : Resource) : int
+getScore() : int
+getFoodgain() : int
+getWorkerCount() : int
+getFreeWorkers() : int
+getColour() : Colour
+getTools() : Tool*
+getString() : QString
+maxToolsReached() : bool
+deleteExtraTool(tool : int) : void
+resetTools() : void
+resetWorkers() : void
+setWorkersOccupied(amount : int) : void
+save() : QJsonObject
+load(json : QJsonObject&) : void
+calcScore() : int
+calcTieBreak() : int
+getLowestToolLevel() : int
+getBuildingCount() : int
+getExtraTools() : std::vector<int>
dataChanged() : void <<signal>>
```