Shotten Totten project report

Baptiste Buvron, Soudarsane TILLAI, ericluo-lab, Marin Bouanchaud, Dashstrom

2023-06-18

SchottenTotten

Gameplay in Schotten Totten resembles simultaneous play of nine separate hands of poker, but where each hand has only three cards in it. There are nine “boundary” stones between players at the start of the game. Players vie to win five of the stones, or three adjacent ones, to win the game.

Download

```
git clone https://github.com/Dashstrom/SchottenTotten
```

Build requirements on Windows

Install Visual Studio Build Tools 2019 from visualstudio.microsoft.com Dont forget to add the “C++ Clang tools for Windows”.

1. Add `C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\VC\Tools\Llvm\x64\bin` to `PATH`
2. Set `VCINSTALLDIR` to `C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\VC`

Download the Windows QT installer at qt.io

1. Make a personal installation and install `Qt6.5 MSVC` version and `CMAKE`.
2. Add `C:\Qt\Tools\CMake_64\bin` and `C:\Qt\6.5.0\msvc2019_64\bin` to your `PATH`

You can now build the project with :

```
cmake -B build -DCMAKE_BUILD_TYPE=Release; cmake --build build --config Release
.\build\Release\SchottenTotten.exe
```

Completion requirements :

Just add `C:\Qt\6.5.0\msvc2019_64\include`, `C:\Qt\6.5.0\msvc2019_64\include\QtCore` and `C:\Qt\6.5.0\msvc2019_64\include\QtWidgets` to your include path in your favorite IDE.

Build requirements on MacOS

Install brew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/
    HEAD/install.sh)"
```

Install `cmake` and `qt6` :

```
brew install cmake qt6
```

Add the following to ~/.zshrc or ~/.bashrc :

```
export PATH="/opt/homebrew/opt/cmake/bin:/opt/homebrew/opt/qt6/bin:$PATH"
```

Build and run with :

```
cmake -B build -DCMAKE_BUILD_TYPE=Release; cmake --build build --config Release
./build/SchottenTotten.app/Contents/MacOS/SchottenTotten
```

Install commit requirements

Setup precommit for autoformat.

```
pip install pre-commit cpplint clang-format
pre-commit install
```

Build README.md as PDF (Linux only)

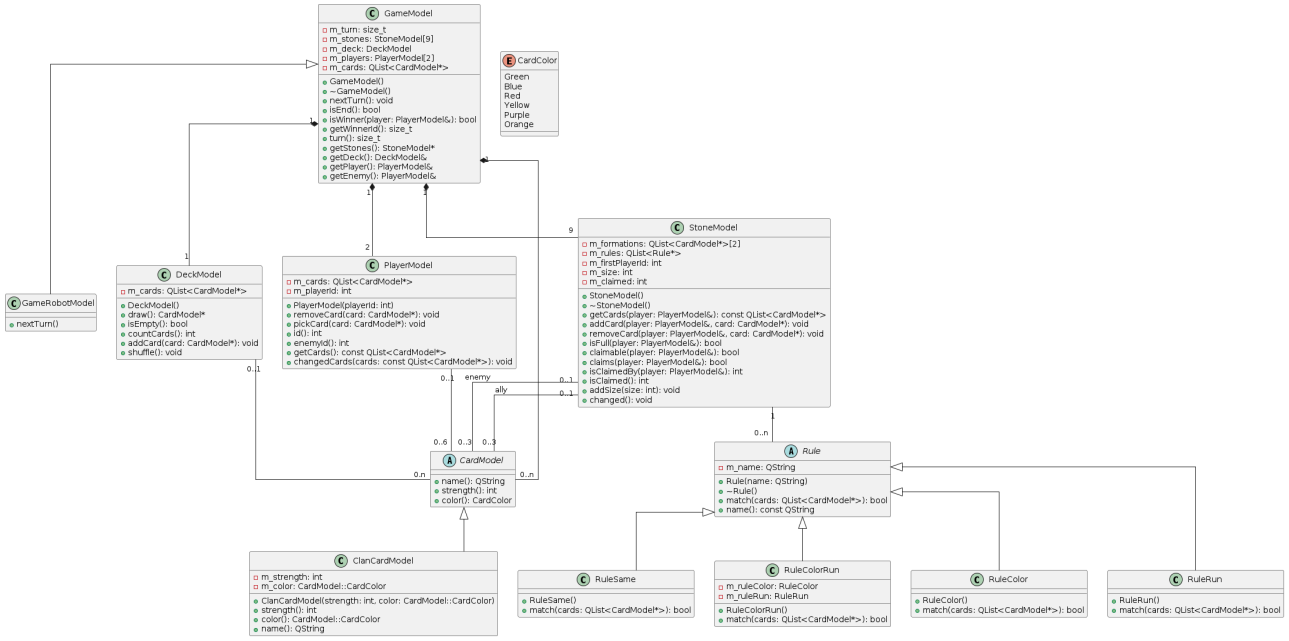
Run the next command and follow the instructions.

```
./docs/build.sh
```

Demo video

You can watch a demo here.

UML



Explanations

We chose to create GameModel and GameRobot Model because it is the view that makes the player play. GameRobotModel plays instead of view and implements an basic random IA.

We created rules on the stones to be able to possibly remove rules later, as could have been the case in the tactical variant.

We chose to make an abstract class for the cards in anticipation of the tactical variant.