

SEP WiSe 2013/ 14

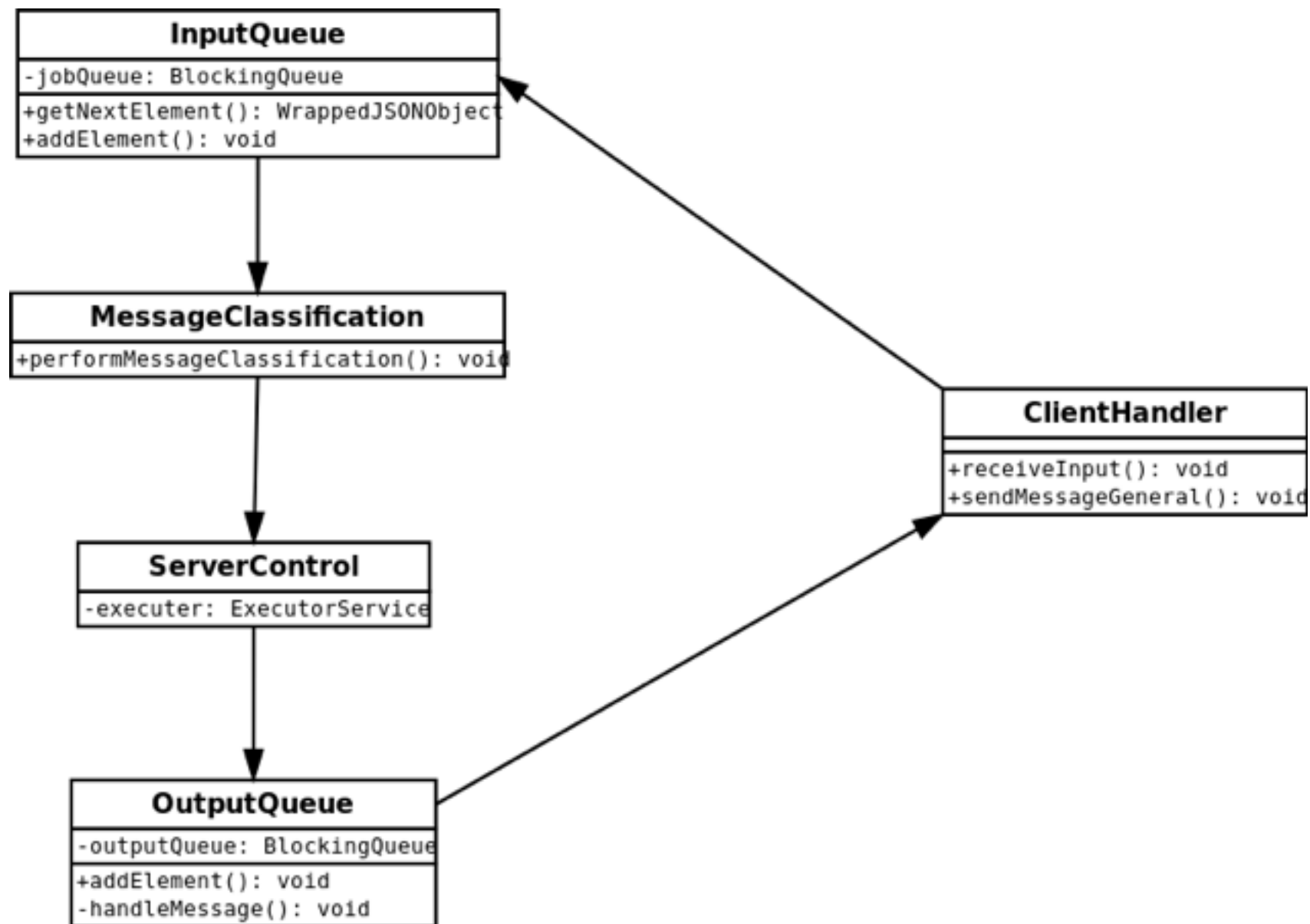
Gruppe Lannister

Message Abarbeitung Server

- Multi-threaded → ClientHandler
- Verwendung von zwei BlockingQueues
 - InputQueue
 - OutputQueue

Message Abarbeitung Server

- Zwei Executor, die die messages bearbeiten
- Executor in ServerControl
 - Abarbeitung incoming messages
- Executor in OutputQueue
 - Senden ausgehender messages



- Vorteile:
 - FIFO → sequentielle Abarbeitung der messages
 - Keine Race Conditions
 - Übersichtliche Struktur
 - Verwendung von Executorn

Kommunikation Observer

- Serverseitige Concurrent LinkedQueue
- Clientseitige Concurrent LinkedQueue



Protokollerweiterung/ Extensions

- "x" : <int>, // x-Koordinate des Spielzugs
"y" : <int>, // y-Koordinate des Spielzugs
"rotation" : < 0 | 1 | 2 | 3 >, // Drehung des Plättchens
- "placement" : <int> // Gebietsindex für zu setzenden Meeple (optional)

Protokollerweiterung/ Extensions

- Wurde erweitert um :
 - “specialMeeple” : <String> // EnumType (Bishop oder Bigmeeple) (optional)
 - Server speichert bei Login Capabilities und zeigt dem Client nur Spiele mit passenden Extensions an

PlayerThread / Serverstabilität

- TileDrawn wird von einem extra Thread bearbeitet
- Eigener virtueller Server war große Hilfe
 - Stabilitätsmaßnahmen
- Null Strings
- Unvollständige Messages
- Falsche Messages

Logik

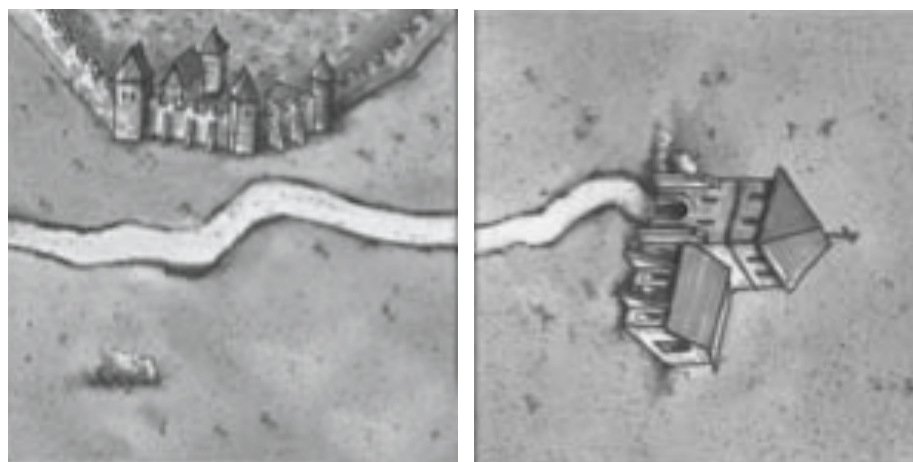
- Areas: [Meadow, Road, Meadow, Town]



- SingleAreas: [Meadow1, Road1, Meadow2, Town]

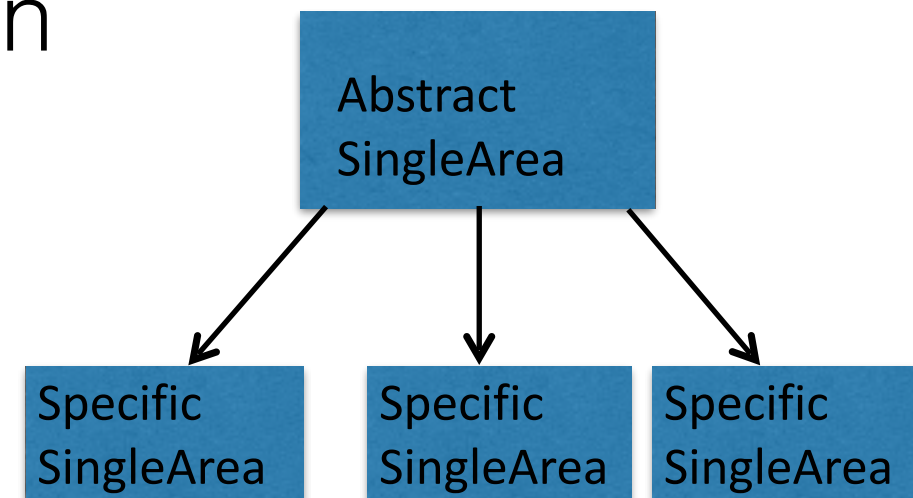


- SingleAreasOnCard1: [M1, R1, M2, T]
- SingleAreasOnCard2: [M3 ,R2, Cloister]
- MergedSingleAreas: [M1, R1, T, Cloister]



- Logik in Klasse „ServerGame“, „Game“ und einzelnen Klassen, welche die verschiedenen SingleAreas repräsentieren.
- Ablauf
 - createInitialCardAndSingleAreas()
 - createSingleAreasForCard(...)
 - mergeEdges(...)

- `List<AbstractSingleArea> singleAreas`
- Kleinere Indizes der Liste entsprechen älterer `SingleArea`
- Neuere Indizes werden beim Merge-Vorgang mit älteren überschrieben



Zugmöglichkeiten der AI

- Einführung der Klasse PossiblePlacement
 - Position
 - Rotation
 - Meeple-Placement
 - Score
 - Priority

- Herausfinden der möglichen Meeple-Platzierungen
- possibleMeeplePlacement - Methode



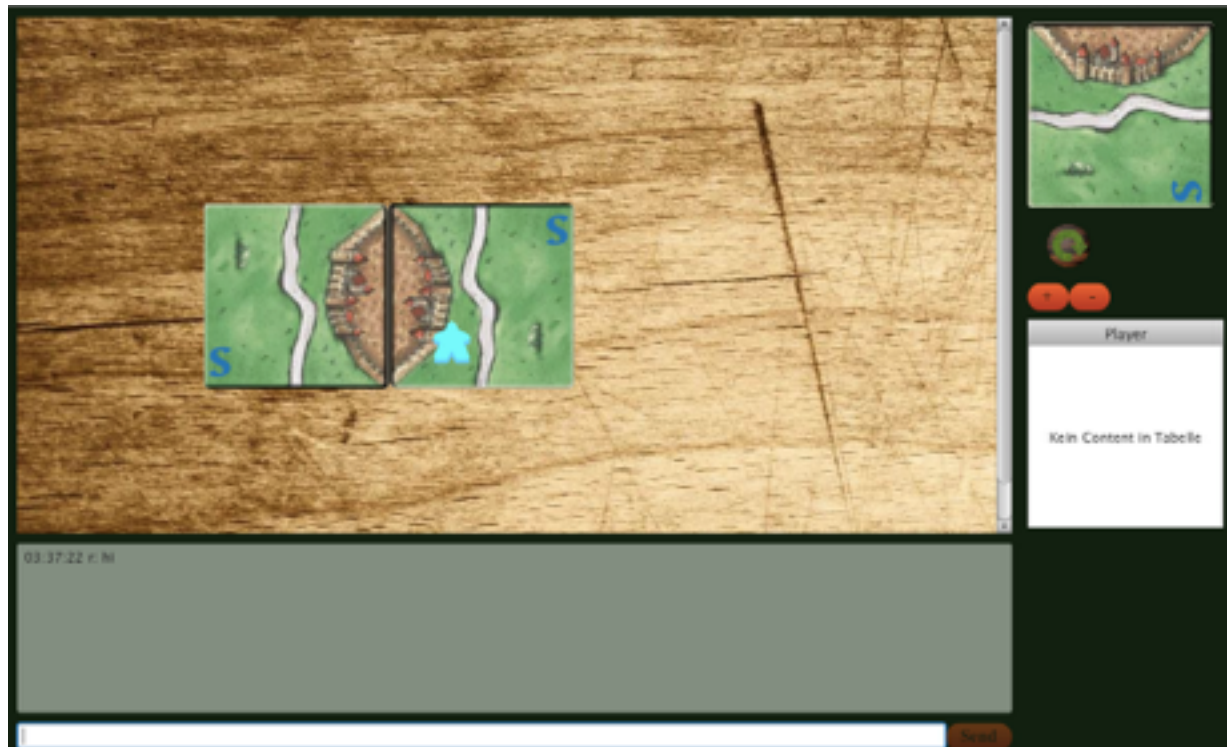
- Herausfinden der möglichen Züge
- checkForLegalPlacements - Methode



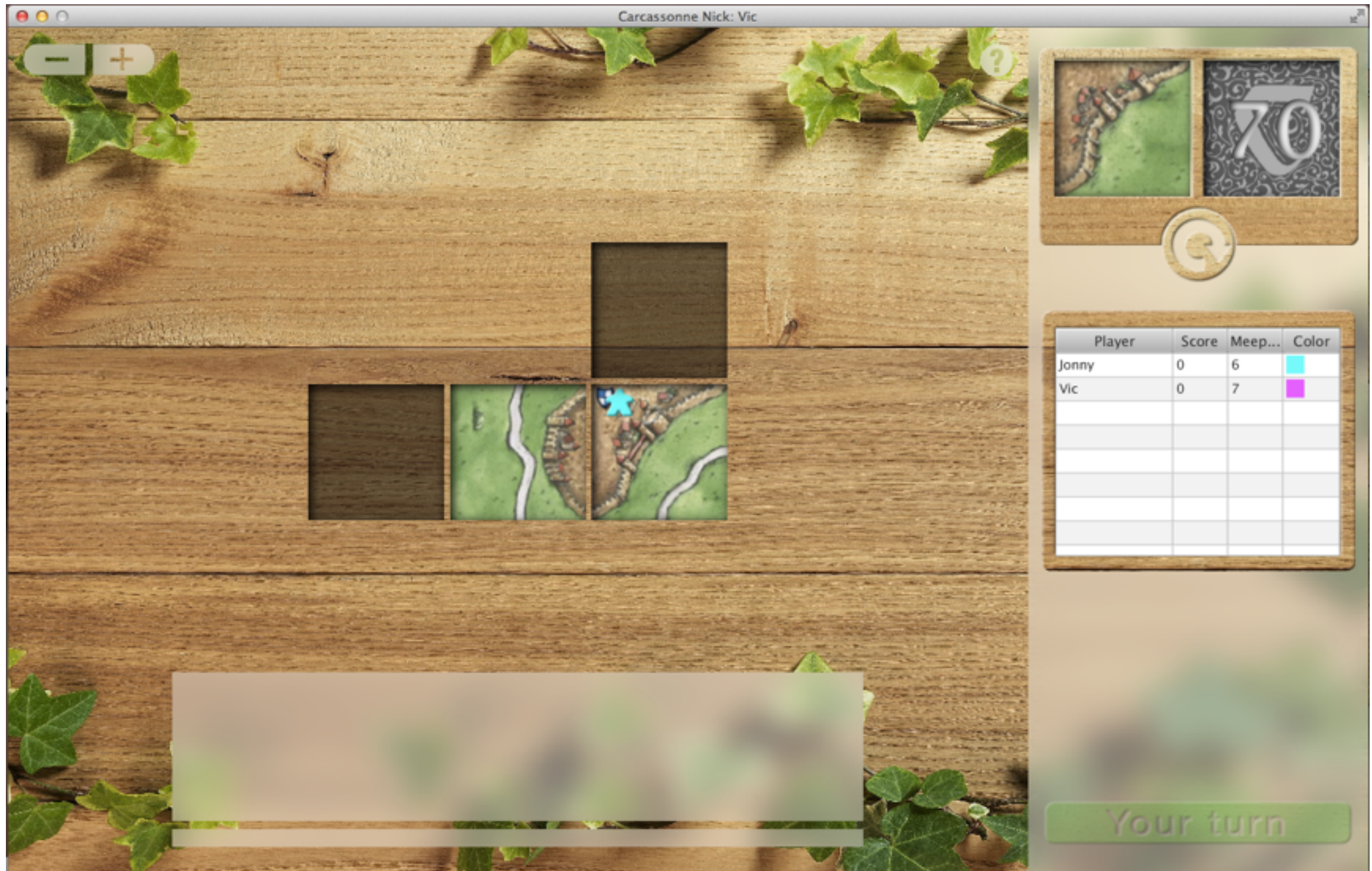
Herausfinden bestmöglicher Zug

- getBestPlacement - Methode
- Überprüfung ob Gebiete abgeschlossen werden können
- wenn nicht, werden die Prioritäten wie folgt gesetzt:
- Kloster erweitern
- Stadt erweitern
- Straße erweitern

GameView



GameView



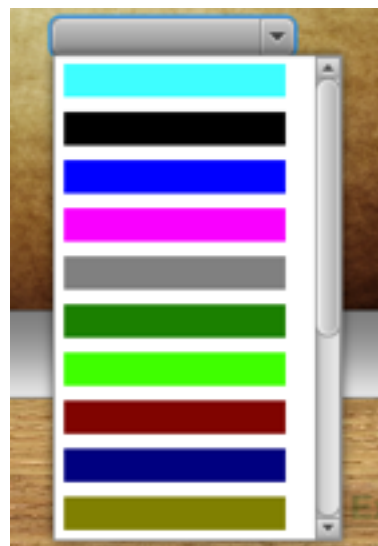
Big Meeple

- Änderung in der Mehrheitsberechnung der Single Areas
- Abfrage ob Big Meeple schon gesetzt wurde

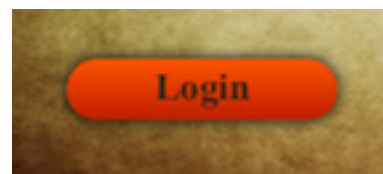
Bishop

- Hinzufügen einer Mehrheitsberechnung für Bischöfe
- SingleAreaMeadow speichert adjazente Klöster in einer weiteren Liste analog zu Städten
- Endbewertung von Städten angepasst
- Schwierigkeit: Darf ich meinen Bischof setzen?

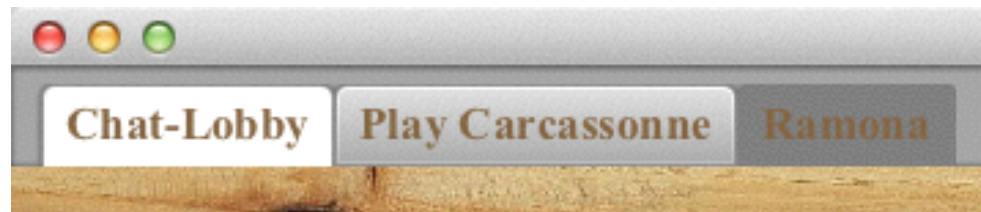
- DropDown Menü Farbe auswählen



- DropShadow für einige Buttons



- Blinkender Tab beim Private Chat



- ServerView
- Grunddesign GameView, ChatLobby, Login

Card Design (inkl. Erweiterungscards)

- Piratendesign



- Friedhofsdesign



- Einzelteile



Meeple Design



Piratenmeeple



Bischofmeeple



Friedhofmeeple



Piraten-Bischofmeeple



Friedhof-Bischofmeeple



Big Meeple