

Softwareengineering Gruppe Poker

3819525 – 6499003 – 7750470 – 5247876

Inhalt

1. Idee
2. Vorgehensmodell
3. Organisation
4. Schwierigkeiten
5. Produkt
 1. Entwicklung
 2. Demo

Idee

- Computeranwendung zum Poker spielen
 - Mit menschlichen Spielern und Bots
- Spielvariante: Texas Holdem
- Spielbar über Konsole

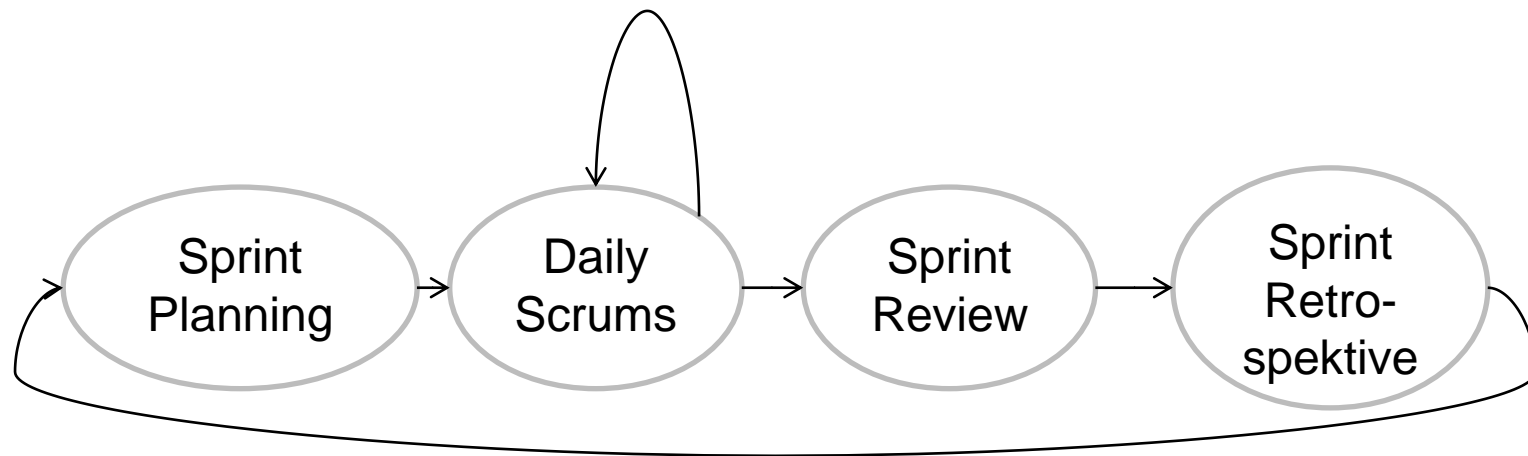
Agiles Vorgehensmodell

- SCRUM
 - Wird in vielen Unternehmen verwendet
 - Automatisch eine gute Struktur

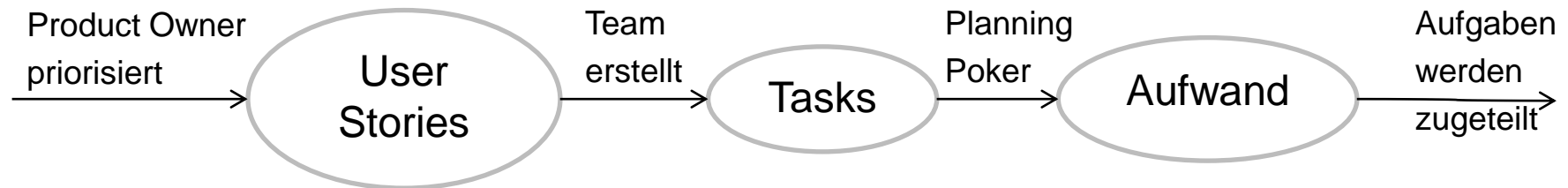
Initiales Scrum Meeting

- Unser Scrum
 - “Daily Scrum” zweimal pro Woche
 - Sprintlänge von 2 Wochen
- Rollenverteilung
 - Scrum Master: Noah Hauke
 - Produkt Owner: Felix Schmitt
 - Team: Alle

Sprint



Sprint Planning



1 Als Spieler möchte ich eine Erklärung der Regeln vor dem Spiel bekommen können, damit ich nicht im Internet suchen muss und damit die Regeln klar definiert sind.

Spielregeln definieren

8

Luca

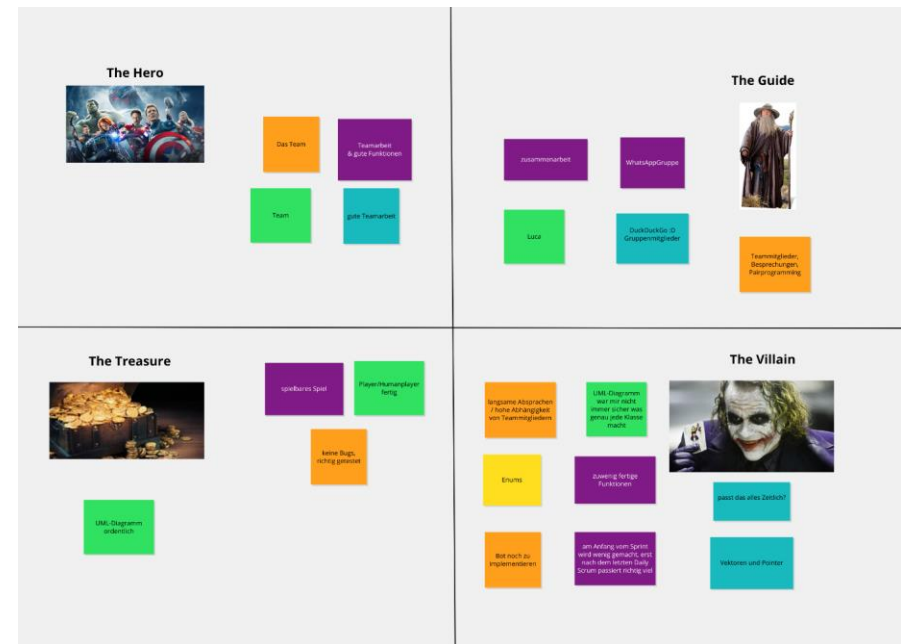
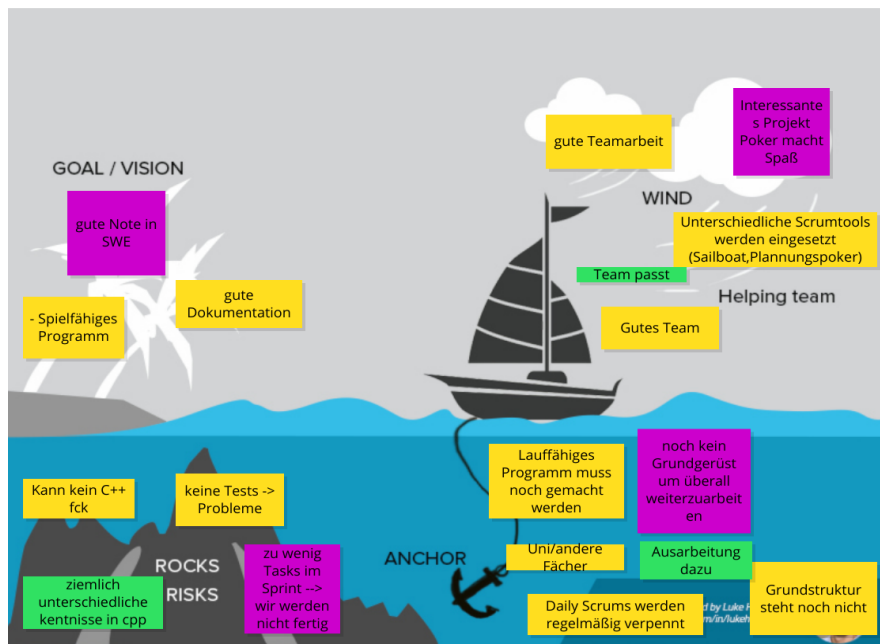
Sprint Review

- Demonstration des Codes oder lauffähigen Programms
- Ableiten von Verbesserungen und in die Sprint Backlog Review Liste eintragen

Sprint Backlog Review Liste 3

- 1 Menü: Die Konsolenausgabe soll nach dem Anzeigen der Regeln wieder geleert werden
- 2 Menü: Schreibfehler beheben
- 3 Menü: Defaultnamen für den Bot anlegen
- 4 Output: Die Ausgabe der Währung soll auch in der Konsole ein \$ darstellen
- 5 Human Player: Reihenfolge der Auswahl ändern
- 6 GameController: Bug in der Spielfunktion (-> Endlosschleife)

Sprint Retrospektive



Sprint Retrospektive









Rezenslon:

Gute Teamarbeit, auch gute Absprachen vor allem bezüglich der Ziele des Projektes. Jedoch wieder die meiste Arbeit gegen Ende des Sprints. Die meisten Ziele des Sprints wurden erreicht, eine Task wurde wegen Zeitmangel aber weggelassen, nächstes Mal wäre es besser wenn das früher erkannt werden würde.



Organisation




- Organisation des Entwicklerteams
 - Microsoft Teams
 - Meetings
 - Kanban Board
 - Issue Liste


Problem ▾	Problembeschreibu... ▾	Priorität ▾	Status ▾	Zugewiesen an ▾	Datum der Meldung ▾	Tage alt ▾
Rules	Ace startet nicht den kleinsten Straight, der wird nicht erkannt		Erledigt	 Rohmann Luca (inf1	07.11.2020	5
Rules	Ein RoyalFlush setzt nicht automatisch StraightFlush bzw. Flush bzw. Straight auf true		Erledigt	 Haas Miriam (inf19'	07.11.2020	3
Menue	ConsoleClear, nach dem anzeigen der Rules		Erledigt	 Schmoll Sophia (inf	05.11.2020	5
Menue	Schreibfehler bei Tischminimum		Erledigt	 Schmoll Sophia (inf	05.11.2020	5
Menue	Für dummen Bot im Menue einen defaultnamen mitgeben		Erledigt	 Schmoll Sophia (inf	05.11.2020	5
Human Player	Reihenfolge der Auswahl ändern; \n nach How much do u want to raise; u -> you		Erledigt	 Hauke Noah (inf191	05.11.2020	7

Issue Liste


Organisation

- Organisation des Entwicklerteams
 - Microsoft Teams
 - Meetings
 - Kanban Board
 - Issue Liste
 - GitHub
 - Versionsverwaltung

 master
  8 branches
  0 tags
 [Go to file](#)
[Add file](#)
[Code](#)

 43-50ph14 Merge pull request #32 from LucRome/Miriam
 3b499b6 38 minutes ago
🕒 262 commits

📁 .vs/SoftwareEngineering/v16/ipch...	idk	8 days ago
📁 Projekt	Systemtest entfernt + FourOfAKind geändert	17 hours ago
📄 .gitignore	weitere RulesTests und erweitertes gitignore	6 days ago
📄 PlannerBoard.PNG	Dokumentation auf aktuellem Stand	5 days ago
📄 Product_Backlog.xlsx	Dokumentation mit Tabellen	5 days ago
📄 README.md	Create README.md	39 minutes ago
📄 Screen1.PNG	Dokumentation auf aktuellem Stand	5 days ago
📄 Software_Engineering_Dokument...	DokuÄnderung	yesterday
📄 Software_Engineering_Projekt_Sc...	Doku	6 days ago
📄 Vorlage_Dokumentation_ideal.docx	Dokumentation	2 months ago

README.md 

Software Engineering

- Projekt für Software Engineering I.
- Realisierung einer Computeranwendung zum Pokerspielen, in der Spielvariante Texas Holdem.
- Die Anwendung soll in der Programmiersprache C++ implementiert werden.
- Die Anwendung soll aus einem einfachen Spielmodus bestehen.
- Mögliche Erweiterungen dazu sind verschiedene Bots, eine graphische Oberfläche oder weitere Spielvarianten von Poker.

Schwierigkeiten

- Einige Tasks waren bis zum Sprintende nicht fertig
- Nicht funktionsfähiger Code auf dem Master-Branch


Abhilfen:

- Mehrere Branches

Default branch

✓ master	an hour ago
----------	-------------

Recent branches

 Miriam	9 hours ago
 Luca	18 hours ago
 Sophia	a day ago
 luca_2	2 days ago
 origin/Noah	4 days ago

Schwierigkeiten

- Einige Tasks waren bis zum Sprintende nicht fertig
- Nicht funktionsfähiger Code auf dem Master-Branch

Abhilfen:

- Mehrere Branches
- Definition of Done

Definition of Done



Code programmieren



Code compilieren



Pull Request anfordern (Code Review)



Merge mit dem Masterbranch

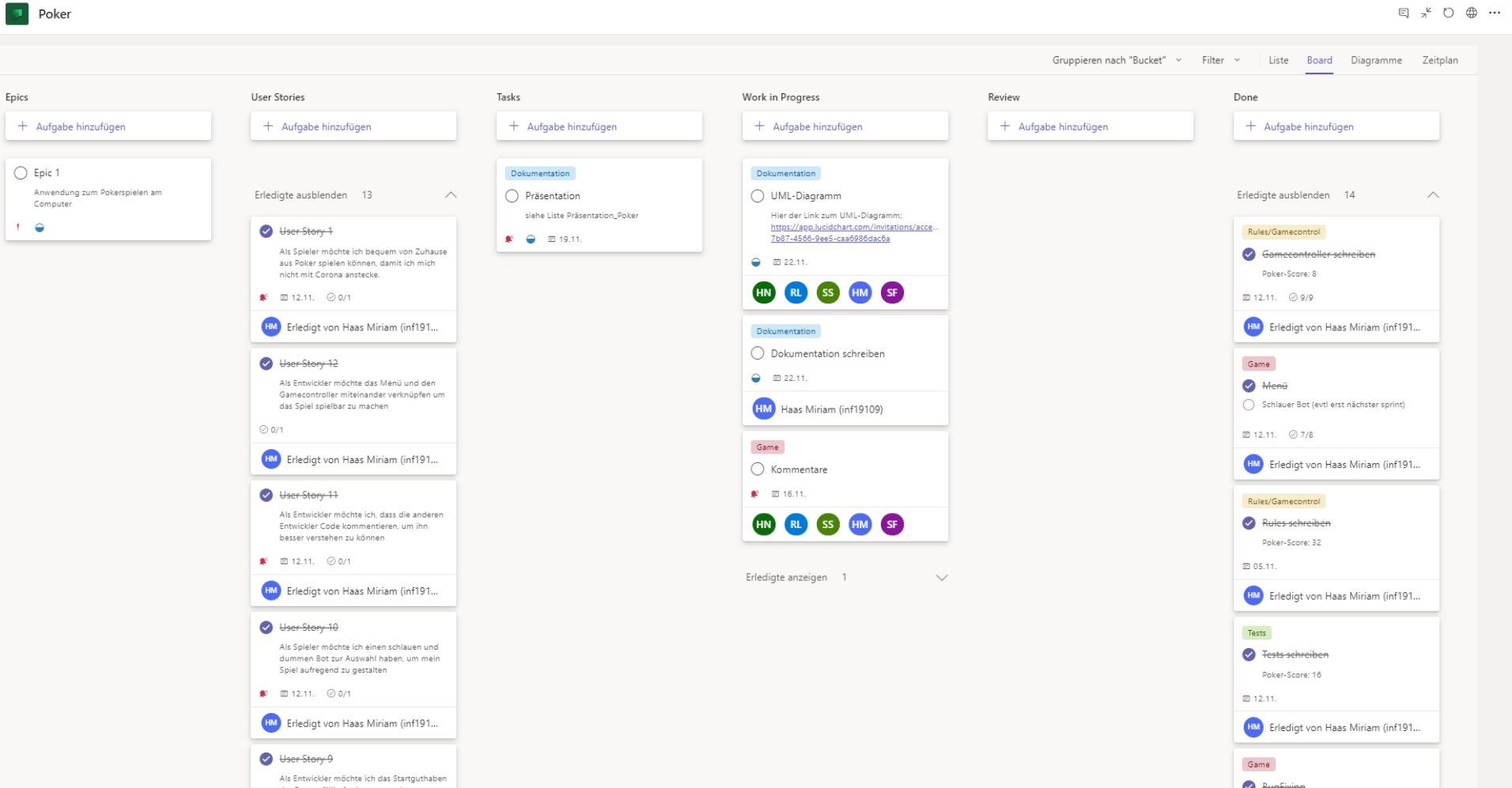
Definition of Done

Schwierigkeiten

- Einige Tasks waren bis zum Sprintende nicht fertig
- Nicht funktionsfähiger Code auf dem Master-Branch

Abhilfen:

- Mehrere Branches
- Definition of Done
- Kanban-Board



Das Kanban-Board

Schwierigkeiten

- Task werden oft erst zum Sprintende begonnen
- Code anderer wird nicht immer direkt verstanden
- Manche Branches sind nicht auf dem aktuellen Stand

Abhilfen:

- Code kommentieren
- GitHub Kenntnisse erweitern

Schwierigkeiten

- Offener Reden, damit Probleme direkt erkannt werden
- Mehr Review in DailyScrum
- PullRequests genauer anschauen

Abhilfen:

- Im nächsten Projekt

Organisationsfazit

- viel Kommunikation
 - gute Absprachen
- Einhaltung von Zeitplänen
- Gruppendynamik
- Überangebot an Softwaretools
- Team hat sich geändert
 - Unklare Neuverteilung von Aufgaben

Was wurde aus der Idee?

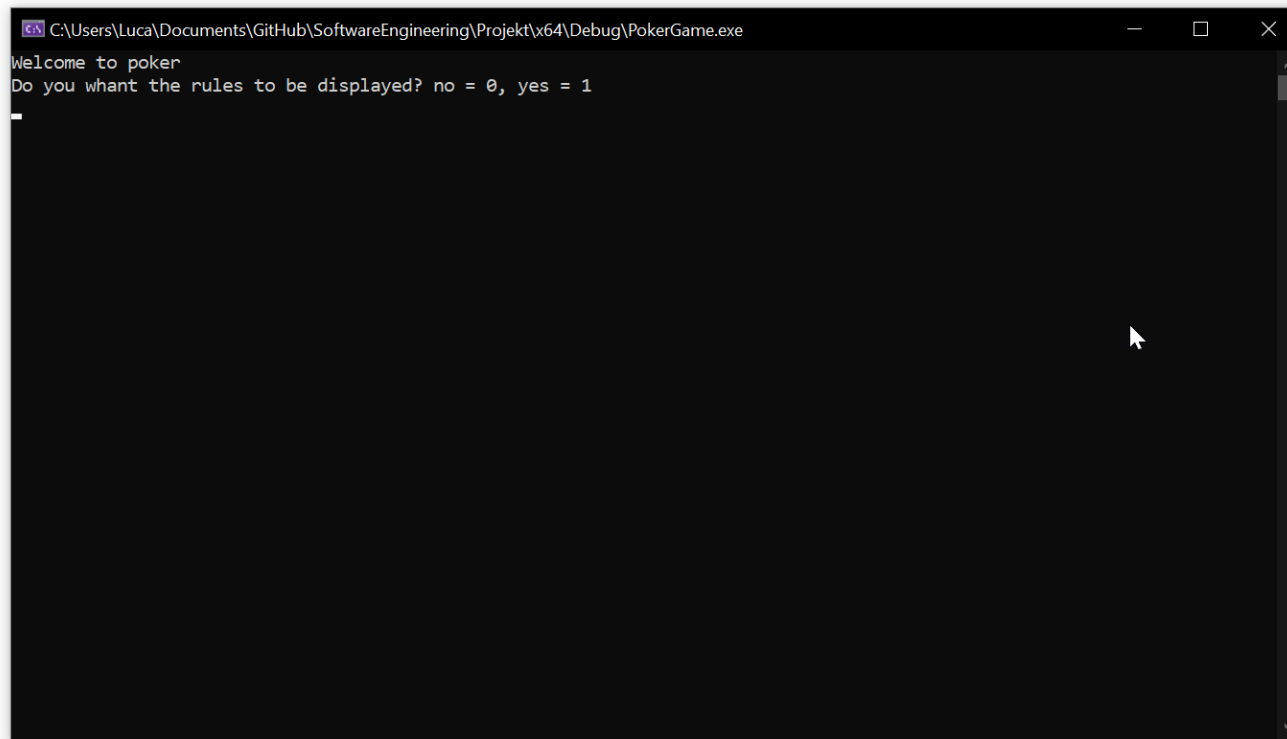
Idee:

- Funktionsfähiges Poker Spiel
- Menschliche Spieler und Bots

Davon umgesetzt:

- Alles außer schlauer Bot
- Dafür: Zufallsbasierter Bot

Vorstellung des Spiels

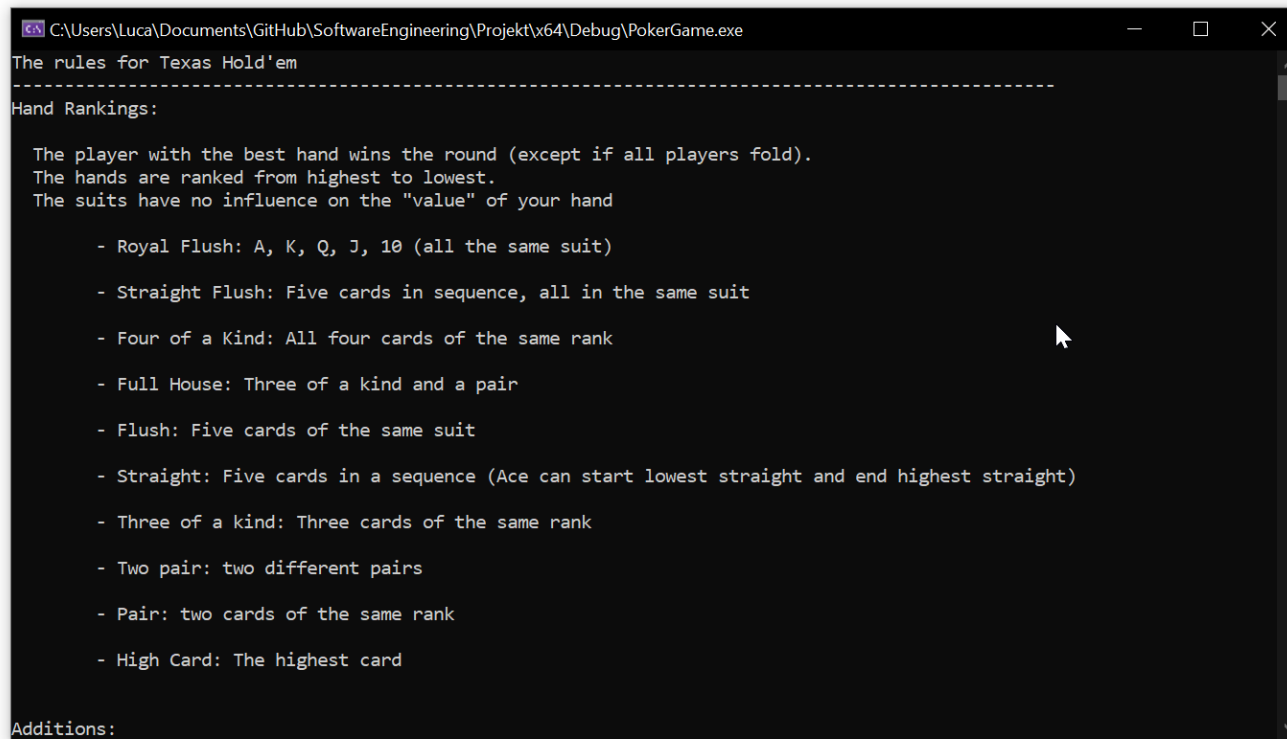


A screenshot of a Windows command prompt window. The title bar shows the file path: `C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe`. The window contains the following text:

```
Welcome to poker  
Do you want the rules to be displayed? no = 0, yes = 1
```

The rest of the window is black, and a mouse cursor is visible on the right side.

Vorstellung des Spiels



```
C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe
The rules for Texas Hold'em
-----
Hand Rankings:

The player with the best hand wins the round (except if all players fold).
The hands are ranked from highest to lowest.
The suits have no influence on the "value" of your hand

- Royal Flush: A, K, Q, J, 10 (all the same suit)
- Straight Flush: Five cards in sequence, all in the same suit
- Four of a Kind: All four cards of the same rank
- Full House: Three of a kind and a pair
- Flush: Five cards of the same suit
- Straight: Five cards in a sequence (Ace can start lowest straight and end highest straight)
- Three of a kind: Three cards of the same rank
- Two pair: two different pairs
- Pair: two cards of the same rank
- High Card: The highest card

Additions:
```

Vorstellung des Spiels

```
C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe
- High Card: Highest card, 2nd highest,...
-----
Roles:
- Dealer: The Dealer hands out the cards (here it is just a role), after each round
          the role is shifted one to the left.
- Small Blind: The player on the left to the dealer has to pay the small blind.
- Big Blind: The player on the left to the small Blind has to pay the big Blind.
-----
Game:
At the end of the round every Player has two pocket Cards, that only the Player can see,
and there are five community cards on the table, that every Player can see.

To build your hand you use up to 4 of the community cards and at least 1 of your pocket cards.
Using only community cards wouldn't make sense, since all players have that hand.
-----
Round:
  Preflop: First the two pocket cards are dealt and there is a round of betting.
  The Flop: (The top card on the stack is thrown away), Three community cards are dealt, round of betting
  The Turn: (The top card on the stack is thrown away), one community card is dealt, round of betting
  The River: (The top card on the stack is thrown away), the last community card is dealt, round of betting
  The Showdown:
```

Vorstellung des Spiels

```
C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe

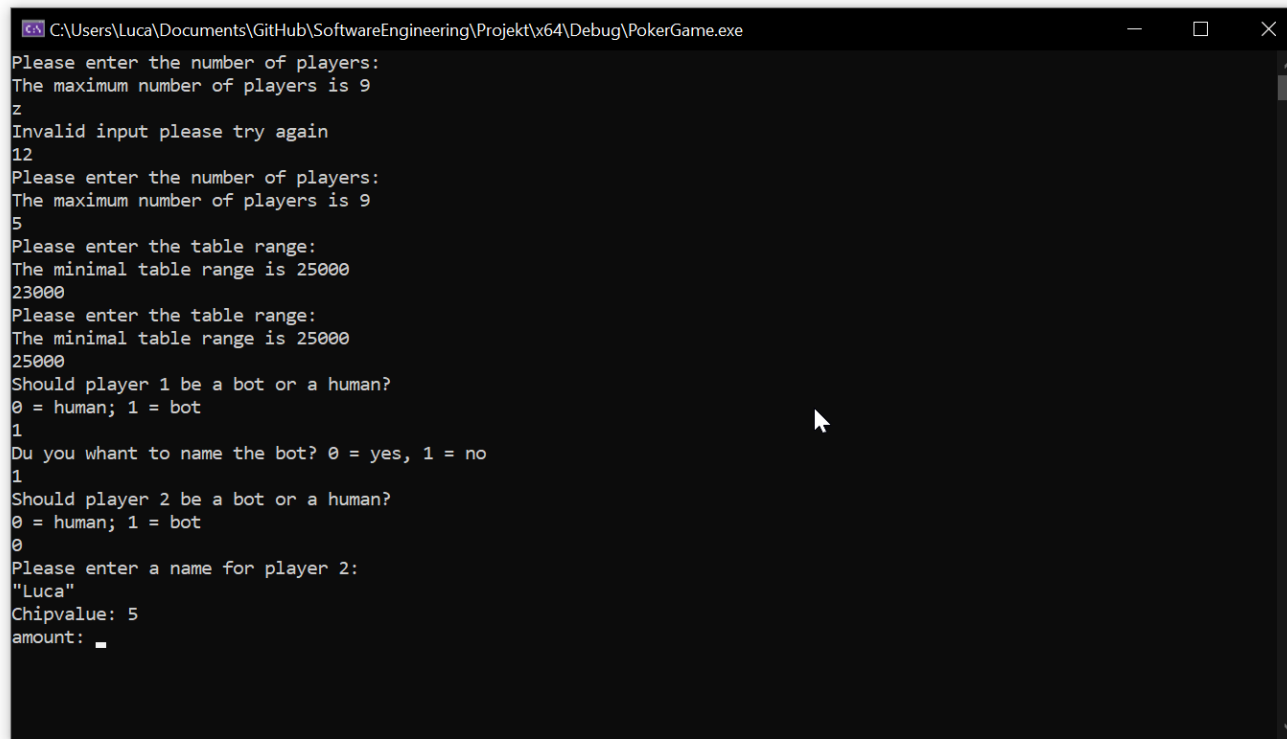
Preflop:
    First the two pocket cards are dealt and there is a round of betting.
The Flop:
    (The top card on the stack is thrown away), Three community cards are dealt, round of betting
The Turn:
    (The top card on the stack is thrown away), one community card is dealt, round of betting
The River:
    (The top card on the stack is thrown away), the last community card is dealt, round of betting
The Showdown:
    The pocket cards are revealed and the winner takes the pot
-----
Round of betting:

    In the Preflop betting round, the first player to act is left of the big blind,
    then this role is always shifted one to the left

    A Player has the following options:
    - Fold: don't pay into the pot and toss away your cards, wait for next round
    - Call/Check: match the amount of the bet/ don't raise
    - Raise: Raise the amount of the bet
    - Forget: All in ()
    - (Begin of Preflop: bet = big blind)

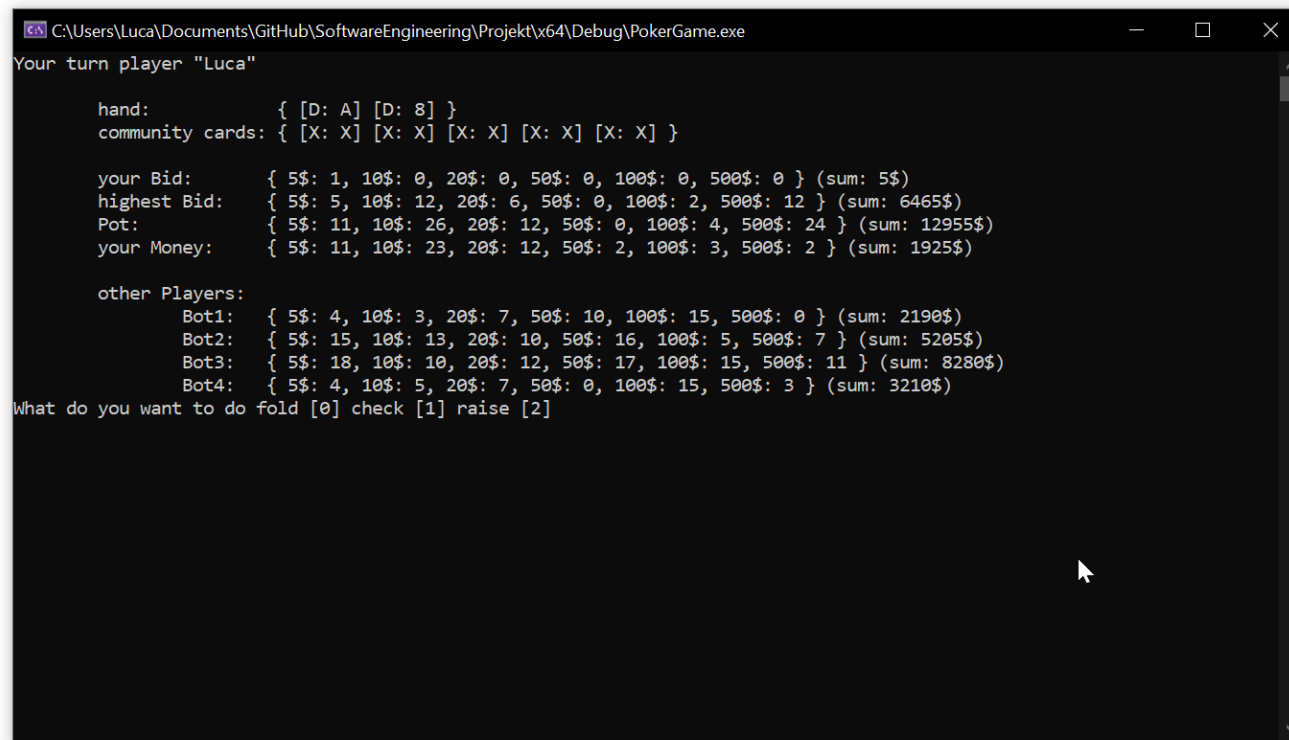
    The round of betting continues until the following conditions are met:
    - all players had a chance to act
    - all players that haven't folded have the same amount of money in the pot
-----
Source:
https://www.pokerlistings.com/poker-rules-texas-holdem
Drücken Sie eine beliebige Taste . . .
```

Vorstellung des Spiels



```
C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe
Please enter the number of players:
The maximum number of players is 9
z
Invalid input please try again
12
Please enter the number of players:
The maximum number of players is 9
5
Please enter the table range:
The minimal table range is 25000
23000
Please enter the table range:
The minimal table range is 25000
25000
Should player 1 be a bot or a human?
0 = human; 1 = bot
1
Du you whant to name the bot? 0 = yes, 1 = no
1
Should player 2 be a bot or a human?
0 = human; 1 = bot
0
Please enter a name for player 2:
"Luca"
Chipvalue: 5
amount: _
```

Vorstellung des Spiels



```
C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe
Your turn player "Luca"

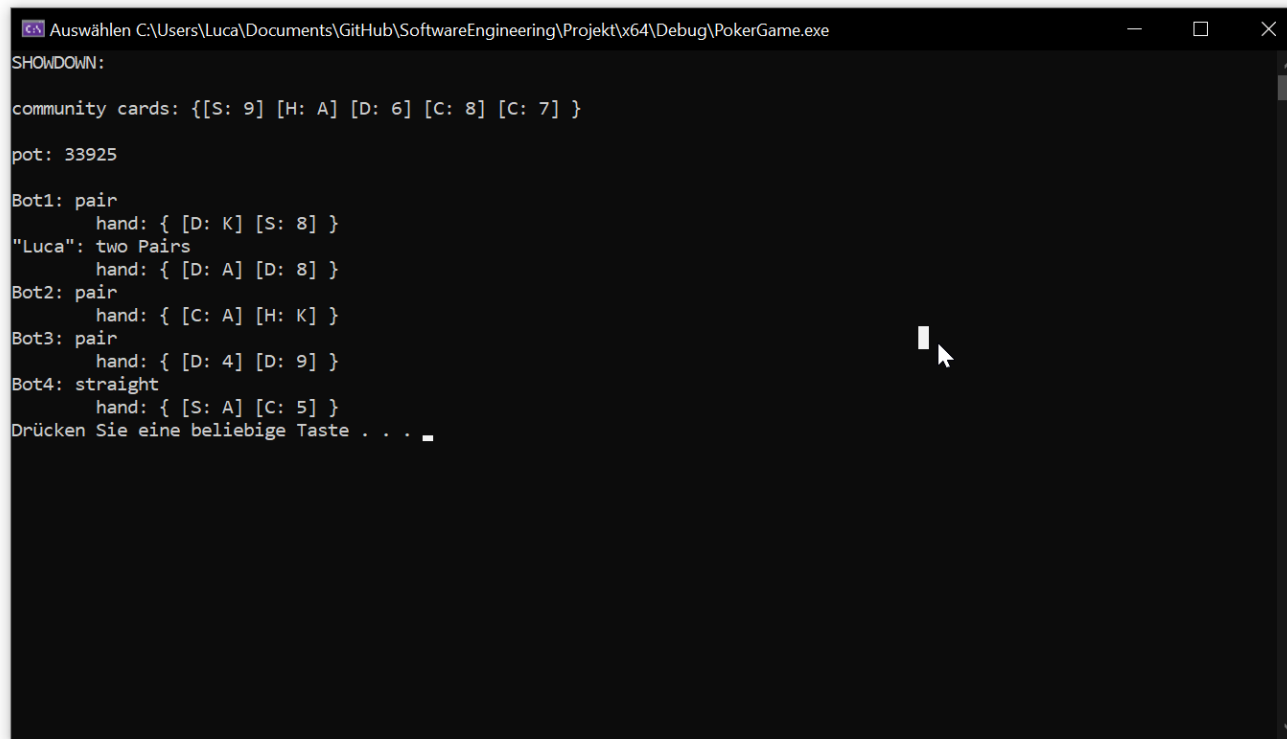
hand:          { [D: A] [D: 8] }
community cards: { [X: X] [X: X] [X: X] [X: X] [X: X] }

your Bid:      { 5$: 1, 10$: 0, 20$: 0, 50$: 0, 100$: 0, 500$: 0 } (sum: 5$)
highest Bid:   { 5$: 5, 10$: 12, 20$: 6, 50$: 0, 100$: 2, 500$: 12 } (sum: 6465$)
Pot:          { 5$: 11, 10$: 26, 20$: 12, 50$: 0, 100$: 4, 500$: 24 } (sum: 12955$)
your Money:    { 5$: 11, 10$: 23, 20$: 12, 50$: 2, 100$: 3, 500$: 2 } (sum: 1925$)

other Players:
  Bot1:        { 5$: 4, 10$: 3, 20$: 7, 50$: 10, 100$: 15, 500$: 0 } (sum: 2190$)
  Bot2:        { 5$: 15, 10$: 13, 20$: 10, 50$: 16, 100$: 5, 500$: 7 } (sum: 5205$)
  Bot3:        { 5$: 18, 10$: 10, 20$: 12, 50$: 17, 100$: 15, 500$: 11 } (sum: 8280$)
  Bot4:        { 5$: 4, 10$: 5, 20$: 7, 50$: 0, 100$: 15, 500$: 3 } (sum: 3210$)

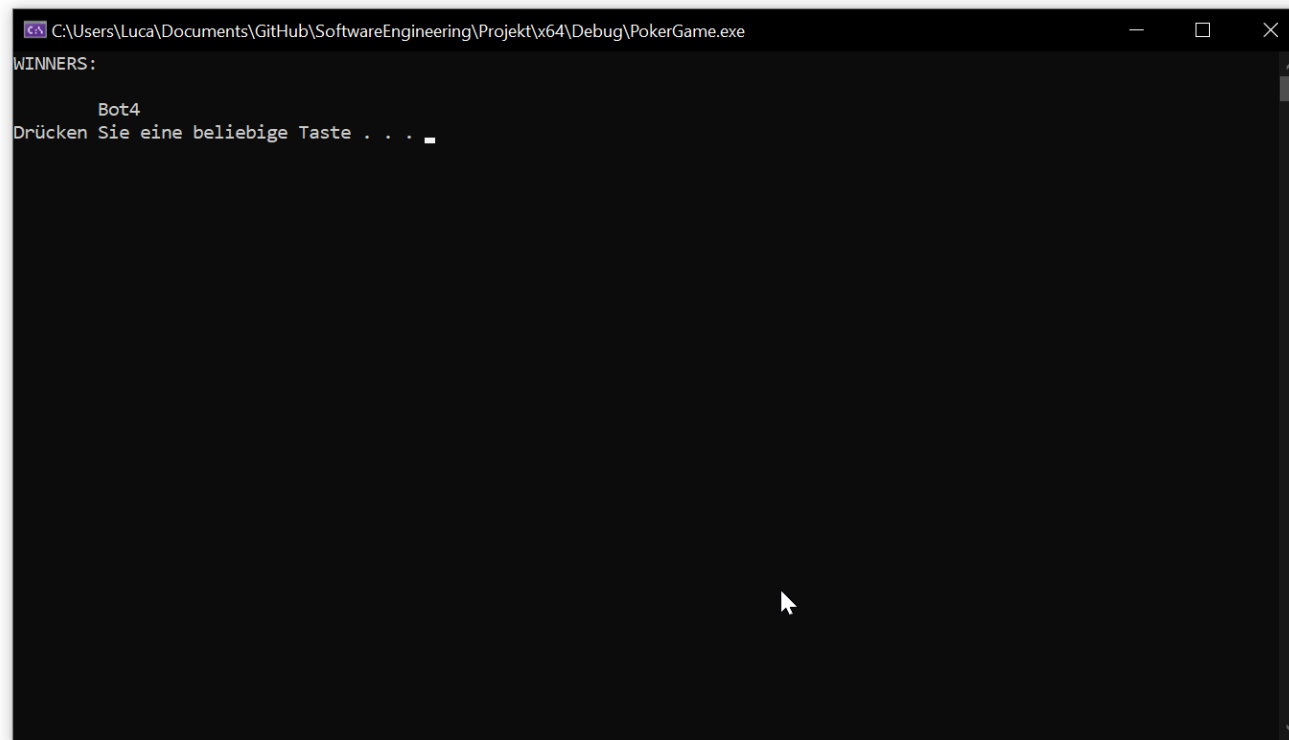
What do you want to do fold [0] check [1] raise [2]
```

Vorstellung des Spiels

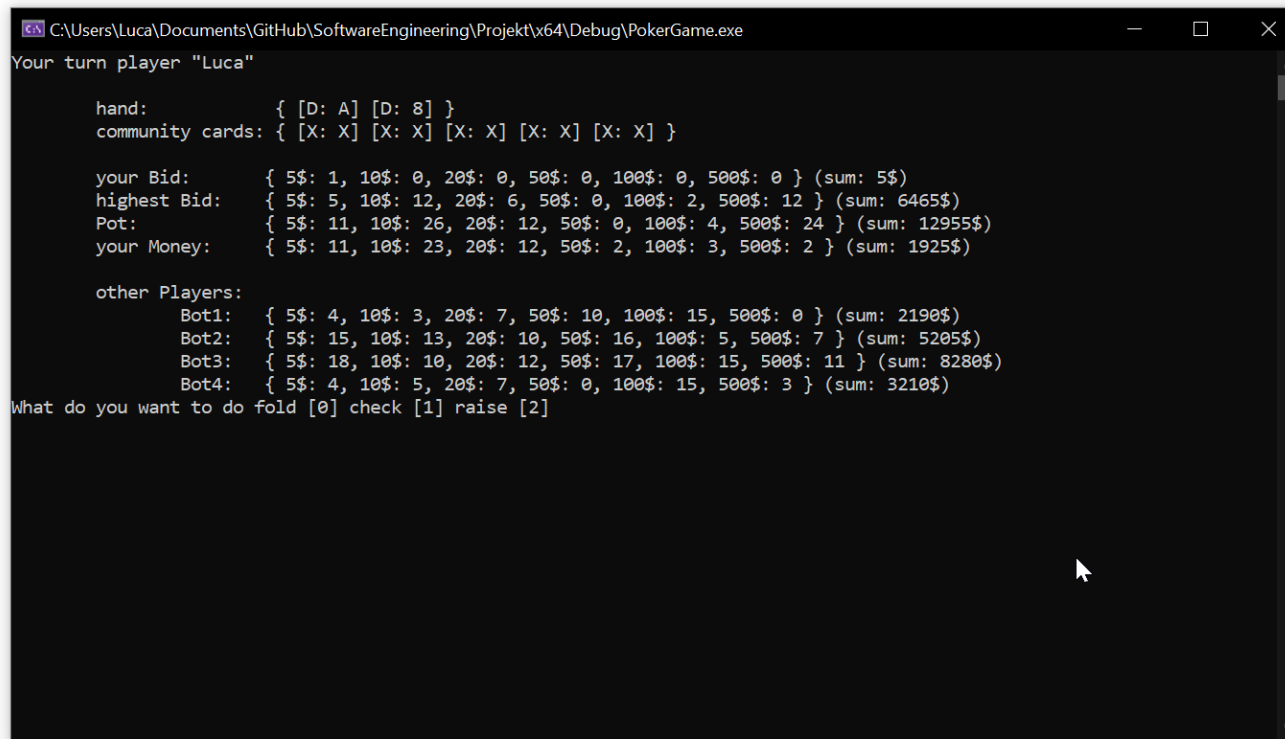


```
Auswählen C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe
SHOWDOWN:
community cards: {[S: 9] [H: A] [D: 6] [C: 8] [C: 7] }
pot: 33925
Bot1: pair
      hand: { [D: K] [S: 8] }
"Luca": two Pairs
      hand: { [D: A] [D: 8] }
Bot2: pair
      hand: { [C: A] [H: K] }
Bot3: pair
      hand: { [D: 4] [D: 9] }
Bot4: straight
      hand: { [S: A] [C: 5] }
Drücken Sie eine beliebige Taste . . .
```

Vorstellung des Spiels



Vorstellung des Spiels



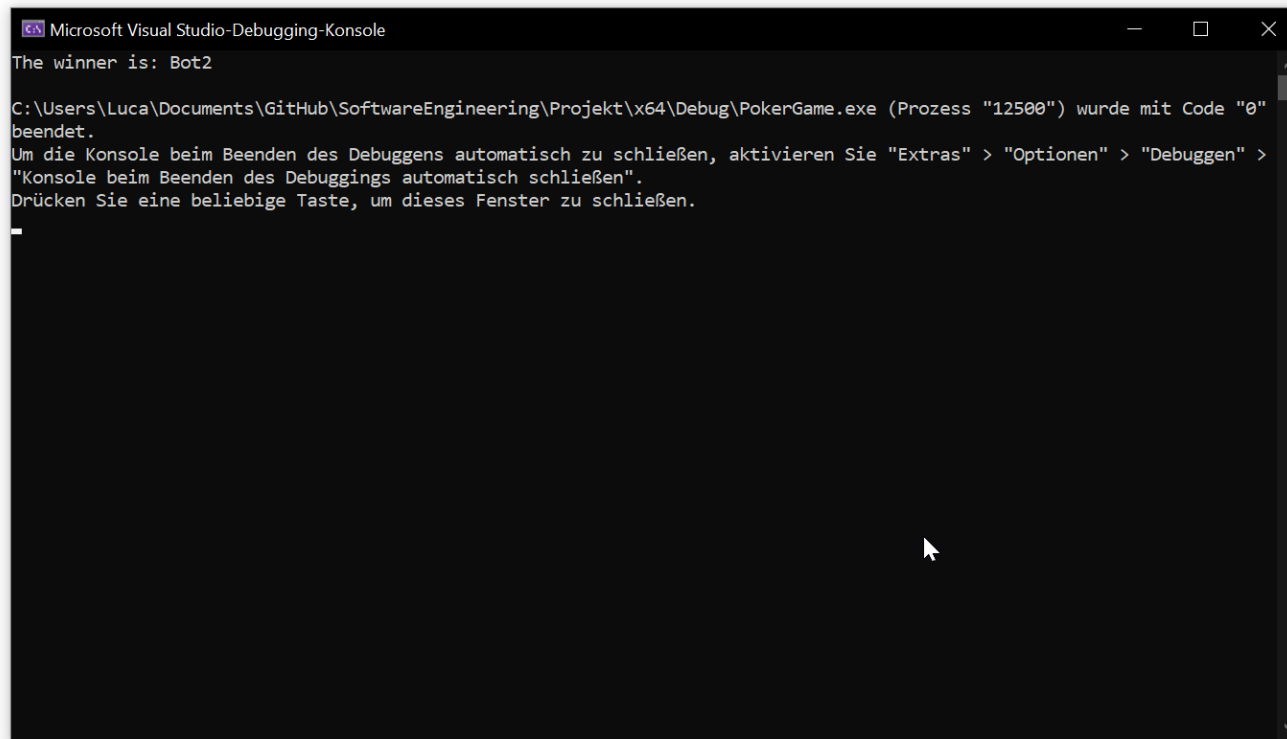
```
C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe
Your turn player "Luca"

hand:          { [D: A] [D: 8] }
community cards: { [X: X] [X: X] [X: X] [X: X] [X: X] }

your Bid:      { 5$: 1, 10$: 0, 20$: 0, 50$: 0, 100$: 0, 500$: 0 } (sum: 5$)
highest Bid:   { 5$: 5, 10$: 12, 20$: 6, 50$: 0, 100$: 2, 500$: 12 } (sum: 6465$)
Pot:          { 5$: 11, 10$: 26, 20$: 12, 50$: 0, 100$: 4, 500$: 24 } (sum: 12955$)
your Money:    { 5$: 11, 10$: 23, 20$: 12, 50$: 2, 100$: 3, 500$: 2 } (sum: 1925$)

other Players:
  Bot1:        { 5$: 4, 10$: 3, 20$: 7, 50$: 10, 100$: 15, 500$: 0 } (sum: 2190$)
  Bot2:        { 5$: 15, 10$: 13, 20$: 10, 50$: 16, 100$: 5, 500$: 7 } (sum: 5205$)
  Bot3:        { 5$: 18, 10$: 10, 20$: 12, 50$: 17, 100$: 15, 500$: 11 } (sum: 8280$)
  Bot4:        { 5$: 4, 10$: 5, 20$: 7, 50$: 0, 100$: 15, 500$: 3 } (sum: 3210$)
What do you want to do fold [0] check [1] raise [2]
```

Vorstellung des Spiels

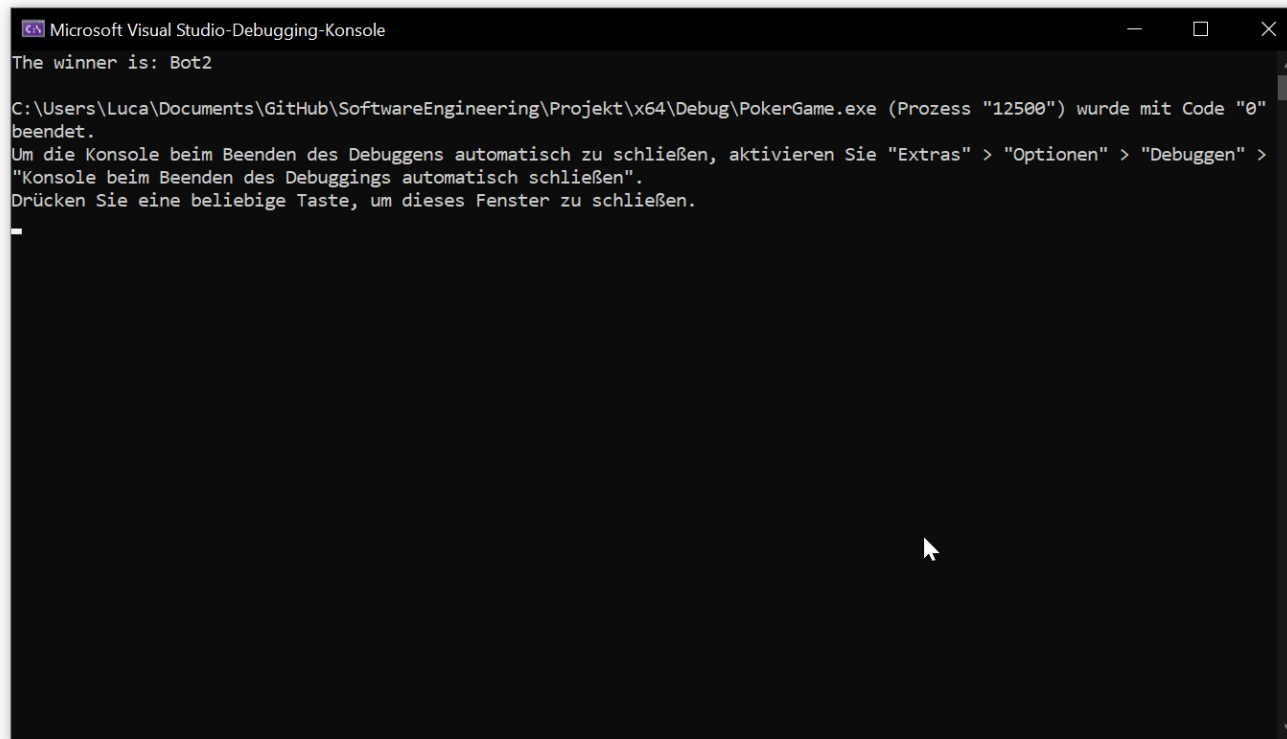


```
Microsoft Visual Studio-Debugging-Konsole

The winner is: Bot2

C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe (Prozess "12500") wurde mit Code "0"
beendet.
Um die Konsole beim Beenden des Debuggens automatisch zu schließen, aktivieren Sie "Extras" > "Optionen" > "Debuggen" >
"Konsole beim Beenden des Debuggings automatisch schließen".
Drücken Sie eine beliebige Taste, um dieses Fenster zu schließen.
```

Vorstellung des Spiels



```
Microsoft Visual Studio-Debugging-Konsole

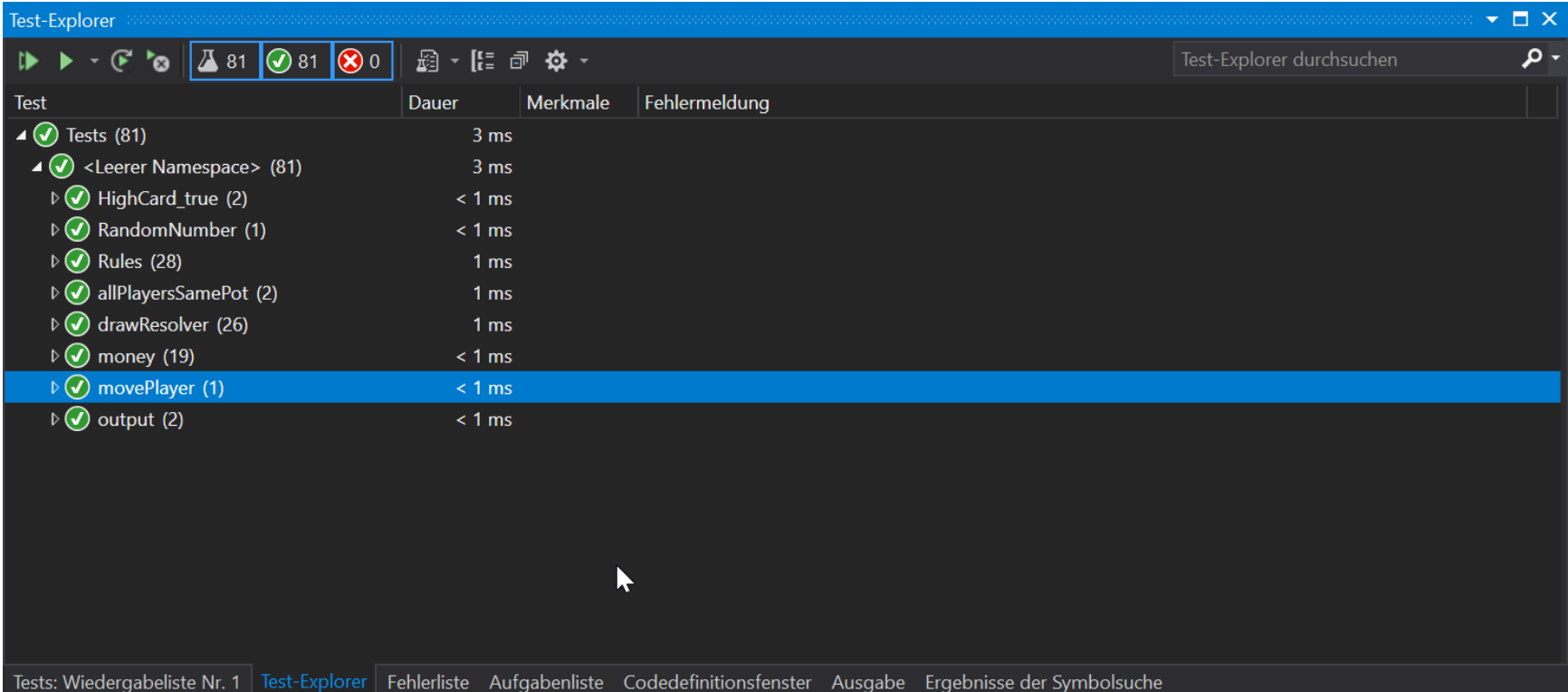
The winner is: Bot2

C:\Users\Luca\Documents\GitHub\SoftwareEngineering\Projekt\x64\Debug\PokerGame.exe (Prozess "12500") wurde mit Code "0"
beendet.
Um die Konsole beim Beenden des Debuggens automatisch zu schließen, aktivieren Sie "Extras" > "Optionen" > "Debuggen" >
"Konsole beim Beenden des Debuggings automatisch schließen".
Drücken Sie eine beliebige Taste, um dieses Fenster zu schließen.
```

Coding Conventions

```
19     private:
20         void round(); //from preflop to showdown
21     void resetAfterRound(); //reset membervariables for a new round
22     //Functions for a round (see Rules.txt)
23     bool preflop(int start_playerNr);
24     bool flop(int start_playerNr);
25     bool turn(int start_playerNr);
26     bool river(int start_playerNr);
27     void showdown(int start_playerNr);
28
29     //bid (after each step)
30     bool roundOfBidding(int start_playerNr);
31     //move of one player
32     plays movePlayer(int playerNr);
33     std::vector<plays> possiblePlays(int playerNr);
34     //have all players bid the same?
35     bool allPlayersSamePot();
36
```

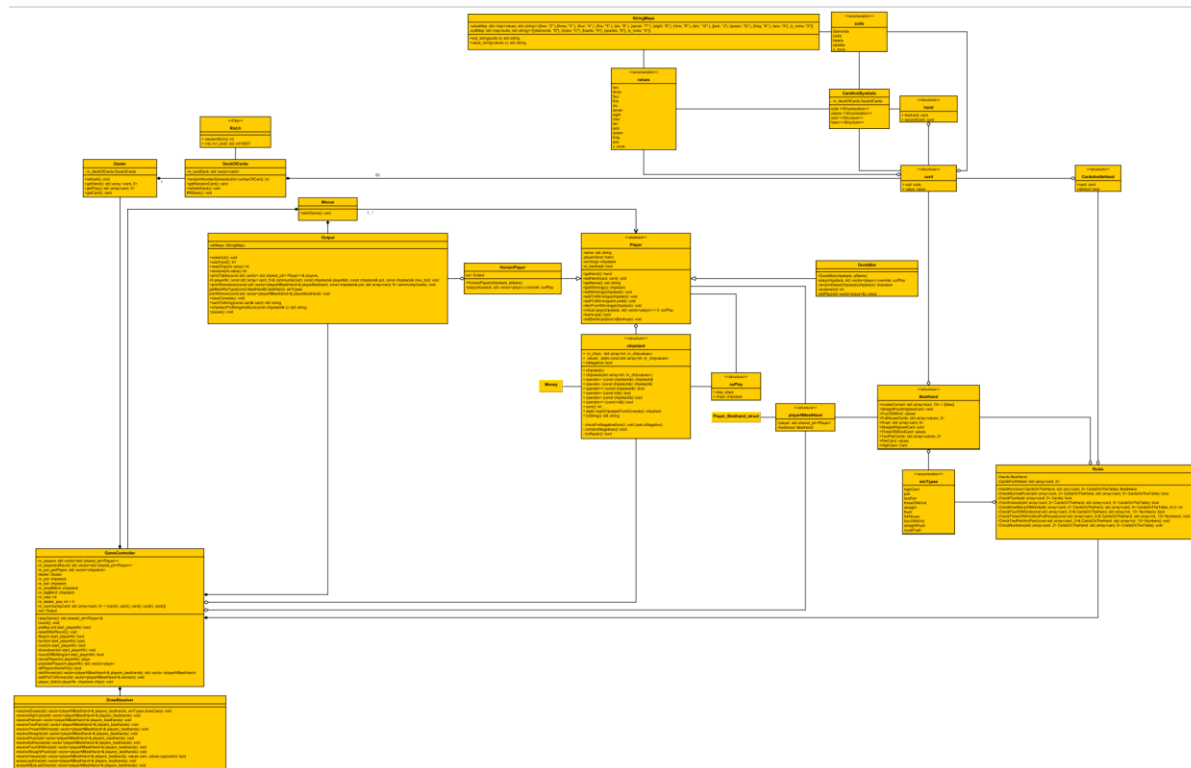
Tests



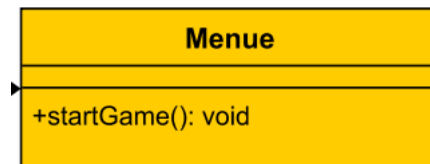
Test	Dauer	Merkmale	Fehlermeldung
Tests (81)	3 ms		
<Leerer Namespace> (81)	3 ms		
HighCard_true (2)	< 1 ms		
RandomNumber (1)	< 1 ms		
Rules (28)	1 ms		
allPlayersSamePot (2)	1 ms		
drawResolver (26)	1 ms		
money (19)	< 1 ms		
movePlayer (1)	< 1 ms		
output (2)	< 1 ms		

Tests: Wiedergabeliste Nr. 1 Test-Explorer Fehlerliste Aufgabenliste Codedefinitionsfenster Ausgabe Ergebnisse der Symbolsuche

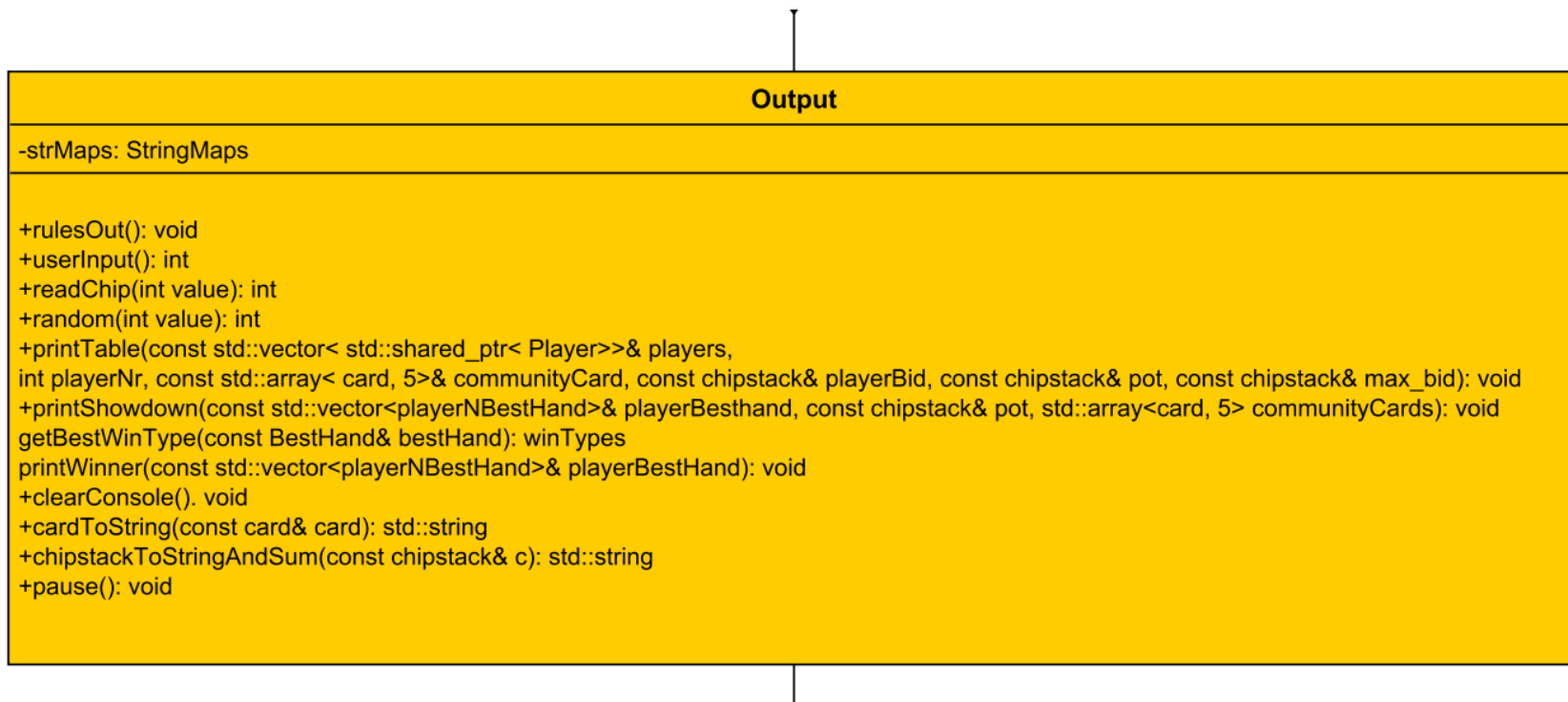
UML



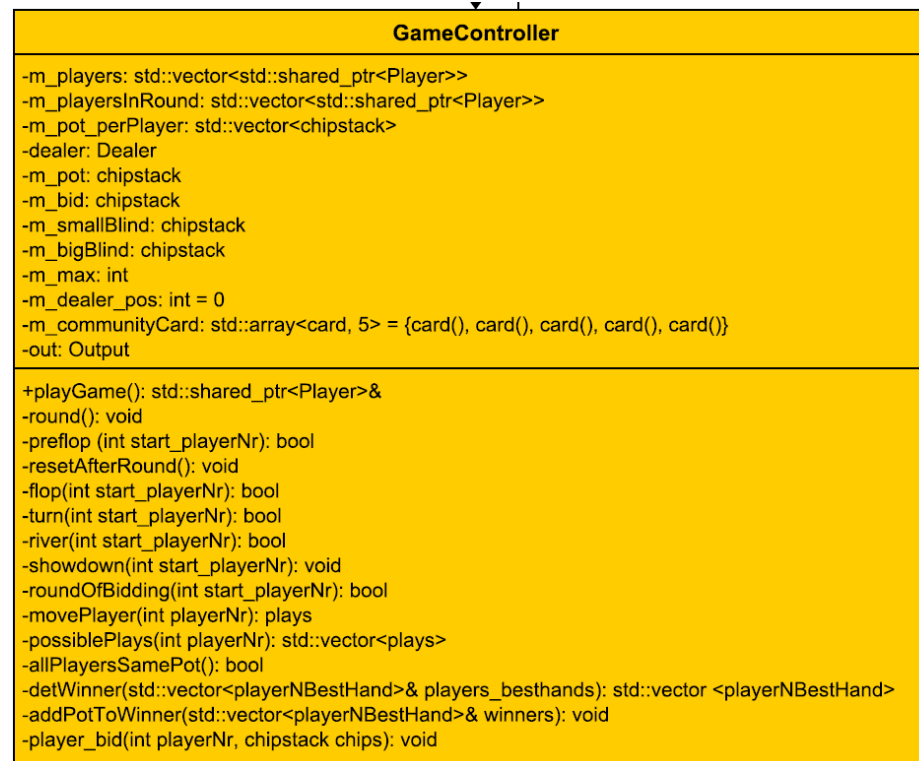
UML



UML



UML



UML

Rules
-Hands BestHand -CardsForAStreet: std::array<card, 5>
+HasWon(hand CardsOnTheHand, std::array<card, 5> CardsOnTheTable): BestHand -CheckNormalFlush(std::array<card, 2> CardsOnTheHand, std::array<card, 5> CardsOnTheTable): bool -CheckFlush(std::array<card, 5> Cards): bool -CheckValues(std::array<card, 2> CardsOnTheHand, std::array<card, 5> CardsOnTheTable): bool -CheckHowManyOfAKind(std::array<card, 2> CardsOnTheHand, std::array<card, 5> CardsOnTheTable, int i): int -CheckFourOfAKind(const std::array<card, 2>& CardsOnTheHand, std::array<int, 13> Numbers): bool -CheckThreeOfAKindAndFullHouse(const std::array<card, 2>& CardsOnTheHand, std::array<int, 13> Numbers): void -CheckTwoPairAndPair(const std::array<card, 2>& CardsOnTheHand, std::array<int, 13> Numbers): void -CheckNumbers(std::array<card, 2> CardsOnTheHand, std::array<card, 5> CardsOnTheTable): void

UML

