

Software Engineering Projekt

Entwicklung einer Anwendung zum Poker spielen am Computer

Informatik
An der Dualen Hochschule Stuttgart

Eingereicht von:

3819525
5247874
6499003
7750470

Studienjahrgang, Kurs:
Bearbeitungszeitraum:

2019, 19IN
10.09.2020 bis 19.11.2020

Inhalt:

1	EINLEITUNG.....	1
1.1	PROJEKTBSCHREIBUNG.....	1
1.2	VORGEHENSMODELL	1
1.3	ORGANISATION	2
1.4	PROGRAMMIERSTANDARDS.....	3
1.5	QUALITÄTSMANAGEMENT	4
1.6	GLEICHBERECHTIGUNG	5
1.7	EPICS UND PRODUCT BACKLOG	5
2	SPRINT 1.....	7
2.1	SPRINT PLANNING AM 24.09.2020	7
2.2	SPRINT REVIEW AM 08.10.2020.....	8
2.3	SPRINT RETROSPEKTIVE AM 08.10.2020	9
3	SPRINT 2.....	10
3.1	SPRINT PLANNING AM 08.10.2020	10
3.2	SPRINT REVIEW AM 22.10.2020.....	11
3.3	SPRINT RETROSPEKTIVE AM 22.10.2020	12
4	SPRINT 3.....	13
4.1	SPRINT PLANNING AM 22.10.2020	13
4.2	SPRINT REVIEW AM 05.11.2020.....	15
4.3	SPRINT RETROSPEKTIVE AM 05.11.2020	15
5	SPRINT 4.....	17
5.1	SPRINT PLANNING 05.11.2020.....	17
5.2	SPRINT-REVIEW.....	18
5.3	SPRINT-RETROSPEKTIVE	19
6	UML DIAGRAMM.....	20
7	ERGEBNIS	21
7.1	PROGRAMM.....	21
7.2	TESTS	24
8	ZUSAMMENFASSUNG.....	26

Abbildungen:

Abbildung 1 PlannerBoard in Teams	2
Abbildung 2 BugFixing Liste	3
Abbildung 3 Definition of Done	4
Abbildung 4 Sailboat Retrospektive	9
Abbildung 5 Helden Retrospektive	12
Abbildung 6 Amazon Produktbewertung Retrospektive.....	16
Abbildung 7 Happiness Histogramm Retrospektive.....	19
Abbildung 8 UML Diagramm.....	20
Abbildung 9 Spielbeginn	21
Abbildung 10 Spielinitialisierung	22
Abbildung 11 GameController	23
Abbildung 12 Erhöhung des Wetteinsatzes	23
Abbildung 13 Showdown	24
Abbildung 14 Winners	24
Abbildung 15 Tests.....	25

Tabellen:

Tabelle 1 Sprint Backlog 1	8
Tabelle 2 Sprint Backlog Review Liste 1	8
Tabelle 3 Verbesserungsliste 1	9
Tabelle 4 Sprint Backlog 2.....	11
Tabelle 5 Sprint Backlog Review Liste 2	11
Tabelle 6 Verbesserungsliste 2	12
Tabelle 7 Sprint Backlog 3.....	14
Tabelle 8 Sprint Backlog Liste 3	15
Tabelle 9 Verbesserungsliste 3	15
Tabelle 10 Sprint Backlog 4.....	18
Tabelle 11 Sprint Backlog Review Liste 4	18
Tabelle 12 Verbesserungsliste 4	19

Abkürzungen:

DoD:	Definition of Done
GUI:	Graphical User Interface
Kartensatz 1:	Besteht aus Karten mit der Gewichtung 1, 2, 3, 5, 8, 13, 25
Kartensatz 2:	Besteht aus Karten mit der Gewichtung 1, 2, 4, 8, 16, 32, 64
UML:	Unified Modeling Language

1 Einleitung

1.1 Projektbeschreibung

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 Spielermodus bestehen. Mögliche Erweiterungen dazu sind verschiedene Bots, eine graphische Oberfläche oder weitere Spielvarianten von Poker.

1.2 Vorgehensmodell

Das Team hat sich einstimmig für ein agiles Vorgehensmodell entschieden. Vor allem aus dem Grund, dass bei der initialen Projektbesprechung noch keine vollumfängliche Beschreibung des Projekts möglich war. Aus dem Kreis der agilen Vorgehensmodelle haben wurde dann konkret für der Scrum-Ansatz ausgewählt. Für diese Entscheidung sprachen vor allem, dass Scrum in vielen Unternehmen als Ansatz für die Softwareentwicklung verwendet wird. Das Team erhoffte sich dadurch erste Erfahrungen mit Scrum zu machen. Außerdem bietet Scrum viele Vorteile durch seine gute Struktur und Einfachheit.

Um diese Projektarbeit und die damit verbundenen Scrumprozesse in den Studienalltag des Teams einbringen zu können, wurden im initialen Projektmeeting die Methode des Process Tailoring angewandt, um das Vorgehensmodell auf die Bedürfnisse des Teams anzupassen.

Festgelegt wurde dabei, dass die DailyScrum Meetings nur zwei Mal die Woche stattfinden sollen. Grund dafür war, dass das Projekt nicht 100% der Zeit beansprucht, und somit auch nicht regelmäßig bzw. täglich am Projekt gearbeitet wird.

Die Sprintlänge wurde auf zwei Wochen initial festgelegt, um auch hier einige Zyklen bis zum Projektende zu durchlaufen, auch wenn dadurch die Größe des Sprint Backlogs kleiner ist als eventuell sinnvoll.

Die Scrum-Rollenverteilung sieht folgende Rollen vor, die vom Team folgendermaßen besetzt wurden:

Scrum Master: Noah Hauke

Product Owner: Felix Schmitt

Die restlichen Projektbeteiligten bilden das Team. In dieses Team sind auch Scrum Master und Product Owner integriert.

1.3 Organisation

Organisiert hat sich das Team über Microsoft Teams. Die Wahl für die erste Besprechung fiel auf Teams, da bei einem Gruppenmitglied Discord nicht funktionierte. Alle weiteren DailyScrums, Sprint Plannings und Sprint Retrospektiven wurden aufgrund dessen auch in Teams abgehalten. Für die Programmierung wurde ein Git-Repository erstellt. Aufgrund keiner wirklichen Kenntnis von Alternativen und da ein Großteil des Teams schon Erfahrungen in GitHub hatte. Dadurch, dass Teams viele weitere Microsoft Produkte wie zum Beispiel Microsoft Planner integriert hat, entschieden sich das Team dazu, sich nicht in eine Software wie Jira einzuarbeiten, sondern in das Team ein Planner Board zu integrieren. Ein Screenshot des Planner Boards ist in Abbildung 1 zu sehen.

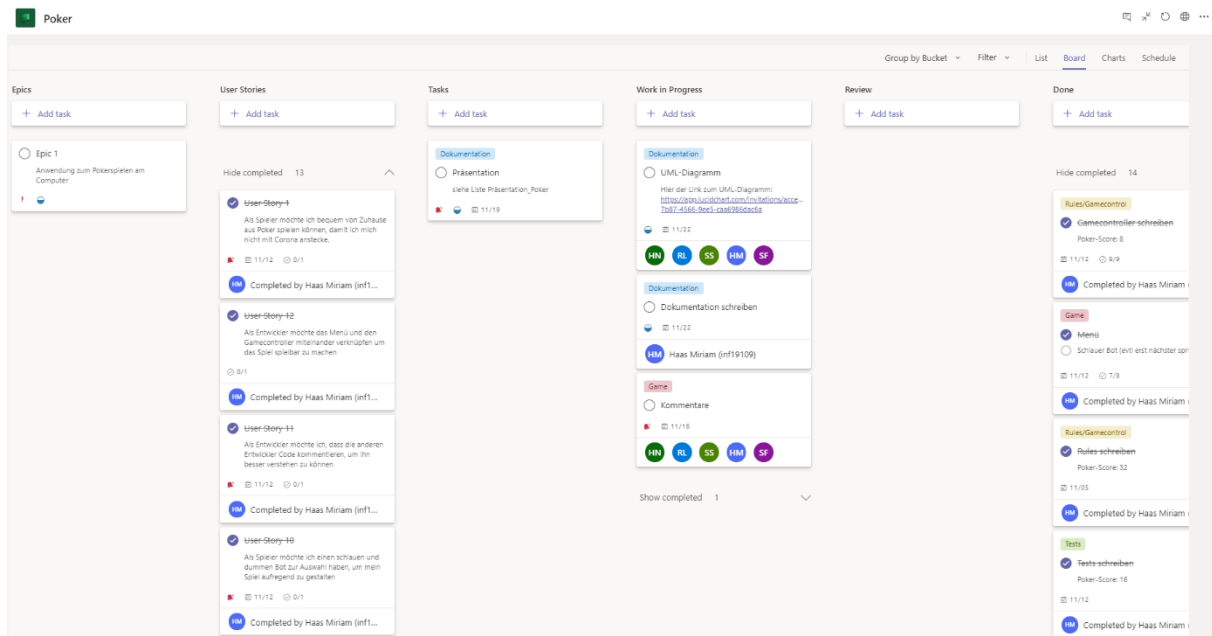
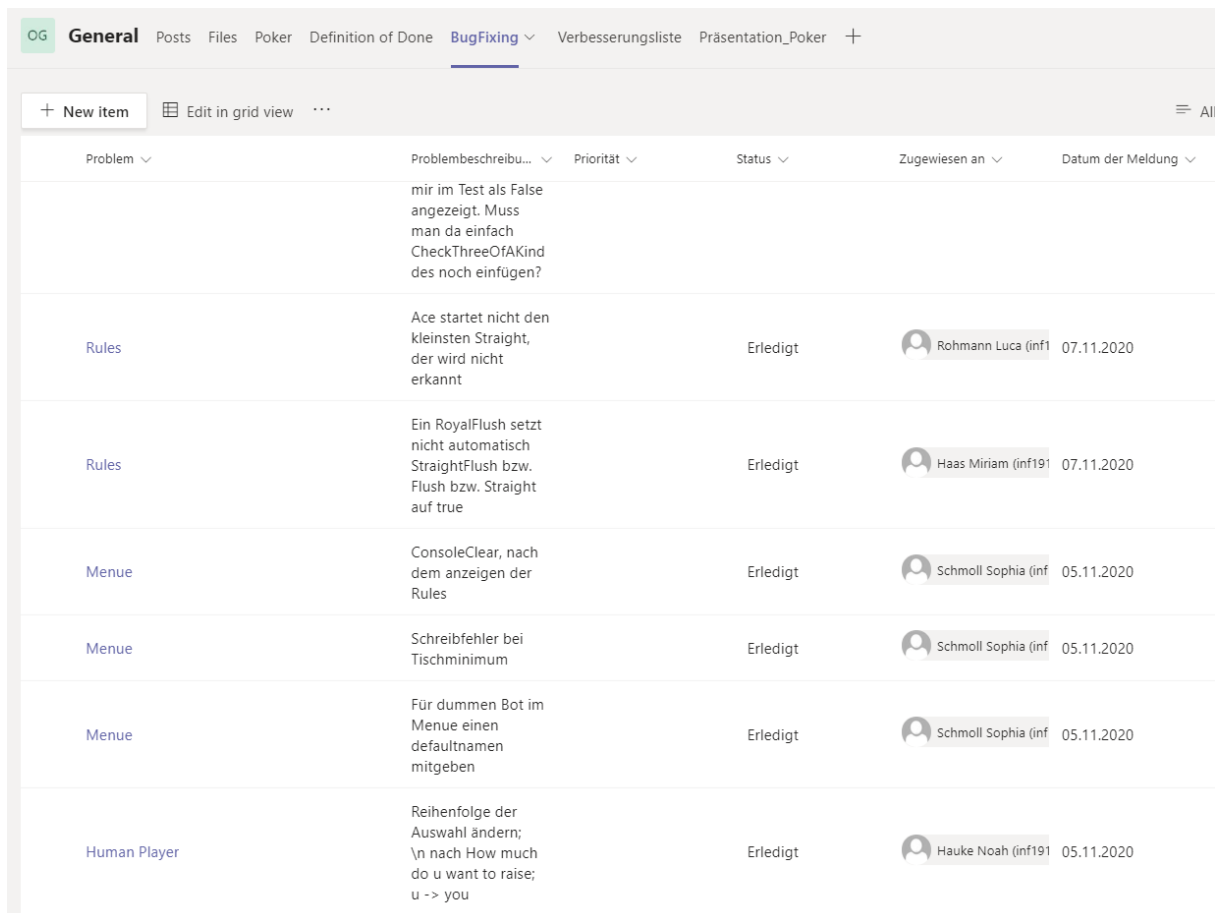


Abbildung 1 PlannerBoard in Teams

Das bot auch den Vorteil, dass man so alles in einer Software halten konnte. Im Laufe des Projektes wurden die Registerkarten immer wieder erweitert, zum einen durch das hinzufügen einer Definition of Done oder einer Registerkarte mit Bugs. Diese erwies sich als nützlich, da dort dann jeder einen Bug und eine potenziell verantwortliche Person eintragen konnte. Zu sehen ist diese Registerkarte in Abbildung 2. Dies hätte auch über Issues in GitHub geregelt werden können, jedoch war dieses Vorgehen auch für die Mitglieder mit geringerer GitHub Erfahrung einfacher.



Problem	Problembeschreibung...	Priorität	Status	Zugewiesen an	Datum der Meldung
	mir im Test als False angezeigt. Muss man da einfach CheckThreeOfAKind des noch einfügen?				
Rules	Ace startet nicht den kleinsten Straight, der wird nicht erkannt		Erledigt	Rohmann Luca (inf1)	07.11.2020
Rules	Ein RoyalFlush setzt nicht automatisch StraightFlush bzw. Flush bzw. Straight auf true		Erledigt	Haas Miriam (inf191)	07.11.2020
Menue	ConsoleClear, nach dem anzeigen der Rules		Erledigt	Schmoll Sophia (inf)	05.11.2020
Menue	Schreibfehler bei Tischminimum		Erledigt	Schmoll Sophia (inf)	05.11.2020
Menue	Für dummen Bot im Menue einen defaultnamen mitgeben		Erledigt	Schmoll Sophia (inf)	05.11.2020
Human Player	Reihenfolge der Auswahl ändern; \n nach How much do u want to raise; u -> you		Erledigt	Hauke Noah (inf191)	05.11.2020

Abbildung 2 BugFixing Liste

1.4 Programmierstandards

Zur Vereinfachung des Prozesses und zur einheitlichen Darstellung der Ergebnisse hat sich das Team auf verschiedene Standards geeinigt.

Die Dateienablage und der Dateienaustausch findet über ein eigens für das Projekt angelegtes Git-Repository statt. Dadurch wird eine konsistente Datenhaltung gewährleistet und ein Datenaustausch kann schnell und einfach vollzogen werden.

Programmiert wird in der Sprache C++, da diese einem Großteil der Teammitglieder bekannt ist.

Als Programmierumgebung wurde Visual Studio ausgewählt, welche bereits bei allen Teammitgliedern installiert war.

Ebenfalls hat sich das Team auf verschiedene Programmierstandards geeinigt:

- Allen Variablen werden englische Namen vergeben
- Alle Variablen haben sprechende Namen
- Ausgabe wird in Englisch verfasst

1.5 Qualitätsmanagement

Im Rahmen der Sprint Retrospektive am 08.10.2020 wurde die Dringlichkeit einer sinnvollen Code-Review vor dem Merge mit dem Masterbranch deutlich. Daher entschied sich das Team dazu, eine Definition of Done zu schreiben, um die Codequalität zu verbessern. Durch die Definition of Done soll vor allem das 4-Augen-Prinzip erfüllt werden. Dabei wird der entwickelte Code von mindestens einem Teammitglied vor der Veröffentlichung bestätigt.

Zudem entschied sich das Team dazu Tests zu schreiben, da das manuelle Testen der Regeln einen erheblichen Zeitaufwand mit sich bringen würde und dieser von keinem Teammitglied geleistet werden kann. Der Zeitaufwand resultiert aus der zufälligen Kartenverteilung und dadurch, dass das Spiel gespielt werden muss und jede Kartenkombination mal aufgetreten sein muss, um von einer Testung aller Funktionen sprechen zu können.

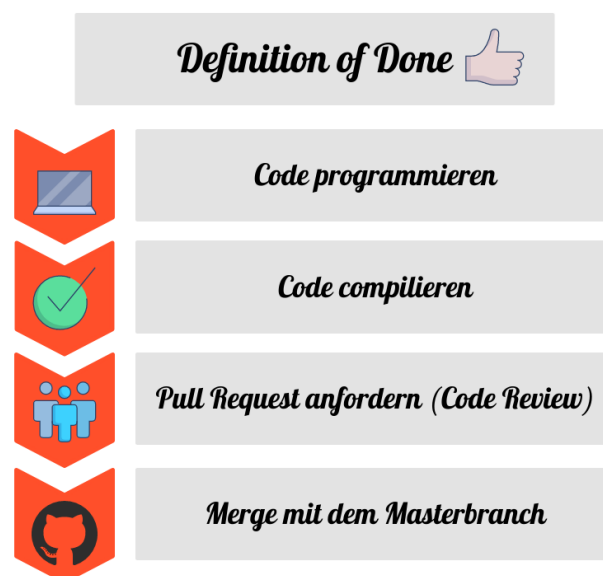


Abbildung 3 Definition of Done

1.6 Gleichberechtigung

Aufgrund der umständlichen und schwer verständlichen Schreibweise zur Gleichberechtigung zwischen Mann und Frau wird in dieser Dokumentation auf die weibliche Form verzichtet. Dies soll aber in keinsten Weise diskriminierend gemeint sein.

1.7 Epics und Product Backlog

Es wurde für das Projekt ein Epic festgelegt:

Epic 1: Anwendung zum Pokerspielen am Computer

Außerdem wurde ein umfassender Product Backlog angelegt, der mit jedem Sprint erweitert wurde:

Nummer	User Story
1	Als Spieler möchte ich bequem von Zuhause aus Poker spielen können, damit ich mich nicht mit Corona anstecke.
2	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.
3	Als Spieler möchte ich die Eingabemöglichkeiten über die Konsole auswählen können, damit das Spiel weiß was ich möchte.
4	Als Entwickler möchte ich das Guthaben der Spieler in Form von Chips darstellen, um das Spiel so realitätsnah wie möglich zu gestalten.
5	Als Spieler möchte ich sowohl gegen reale Player als auch Bots spielen können, um das Spiel abwechslungsreich zu gestalten und um auch mit Freunden/alleine spielen zu können.
6	Als Entwickler möchte ich die Ausgabe in englischer Sprache gestalten, um eine sprachliche Einheit darzustellen und die Ausgabe ordentlich zu gestalten.

-
- 7 Als Spieler möchte ich die Höhe der Blinds und das Tischmaximum festlegen können, um das Spiel an meine Bedürfnisse anpassen zu können.
 - 8 Als Spieler möchte ich die Anzahl der Mitspieler festlegen können, um das Spiel abwechslungsreich machen zu können.
 - 9 Als Entwickler möchte ich das Startguthaben der Bots zufällig festlegen, um ein aufregendes Spiel zu haben.
 - 10 Als Spieler möchte ich einen schlaunen und dummen Bot zur Auswahl haben, um mein Spiel aufregend zu gestalten.
 - 11 Als Entwickler möchte ich, dass die anderen Entwickler Code kommentieren, um ihn besser verstehen zu können.
 - 12 Als Entwickler möchte das Menü und den Gamecontroller miteinander verknüpfen, um das Spiel spielbar zu machen
 - 13 Als Spieler möchte ich das Startguthaben festlegen können, um länger spielen zu können und das Spiel interessanter zu gestalten.

2 Sprint 1

2.1 Sprint Planning am 24.09.2020

Zu Beginn des Meetings wurden vom Product Owner folgende Priorisierung der User Stories vorgenommen. Daraus wurden dann vom Team verschiedene Tasks definiert (siehe Sprint Backlog 1).

Zur Einschätzung der Tasks wurde auf die Methode des Planning Pokers zurückgegriffen. Um dieses durchzuführen, wurde das Online-Tool <https://www.scrumpoker.online> verwendet und der darin implementierte Kartensatz 1.

Dieses Tool erwies sich als sehr nützlich. Damit konnte die Einschätzung der Tasks sehr schnell und effizient vorgenommen werden. Über Uneinigkeiten wurde diskutiert, um auf ein gemeinsames Ergebnis zu kommen.

Der letzte Punkt im Meeting war die Verteilung der einzelnen Tasks auf die Mitglieder des Teams. Dabei wurde sowohl auf eine gleichmäßige Verteilung des Arbeitsvolumens geachtet als auch auf die einzelnen Vorkenntnisse der Teammitglieder.

Sprint Backlog 1

Priorisierung User Story

- 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.
- 2 Als Spieler möchte ich sowohl gegen reale Player als auch Bots spielen können, um das Spiel abwechslungsreich zu gestalten und um auch mit Freunden/alleine spielen zu können.
- 3 Als Spieler möchte ich die Anzahl der Mitspieler festlegen können, um das Spiel abwechslungsreich machen zu können.

Task	Aufwand	Verantwortlich
Grundlegende Architektur	3	Luca
Spielregeln definieren	8	Luca

Spielregeln grob programmieren	8	Felix
Zufallsgenerator	3	Miriam
UML-Diagramm erstellen	13	Noah
Menü aufsetzen	5	Sophia
Spielregeln ausgeben	3	Miriam
Klassen anlegen	5	Miriam

Tabelle 1 Sprint Backlog 1

2.2 Sprint Review am 08.10.2020

Als erster Punkt des Meetings wurde der im letzten Sprint geschriebene Code vorgestellt. Außerdem wurde jedes erstellte Codeelement zusätzlich nach dem 4-Augen-Prinzip allen Teammitgliedern vorgestellt.

Dabei wurden einige Elemente gefunden, die im nächsten Sprint noch bearbeitet werden müssen. Diese sind in der Sprint Backlog Review Liste 1 dargestellt.

Sprint Backlog Review Liste 1

- | | |
|---|--|
| 1 | UML-Diagramm: Änderungen an der Struktur vornehmen |
| 2 | Menü: Zahlendreher auflösen |
| 3 | Rules: Datentypen verändern von String zu Enum |

Tabelle 2 Sprint Backlog Review Liste 1

2.3 Sprint Retrospektive am 08.10.2020

Begonnen wurde das Meeting mit einer Sailboat-Retrospektive, in die jedes Teammitglied seine Gedanken zum vergangenen Sprint eintragen konnte. Aus dieser Grafik wurde dann eine Liste an Verbesserungen für den nächsten Sprint abgeleitet, diese wurden dann in der Verbesserungsliste 1 festgehalten.

Verbesserungsliste 1

- | | | |
|---|---|-------------------------------|
| 1 | Richtiges spielfähiges Programm schreiben bis zum nächsten Sprint | |
| 2 | DailyScrum nicht mehr verpassen | |
| 3 | Tests zur Sprintplanung hinzufügen | Software muss getestet werden |
| 4 | C++-Kenntnisse auf einen Stand bringen | (z.B. durch Pair-Programming) |
| 5 | Projektplanung in Software einbetten | (z.B. Planner von Teams) |
| 6 | Definition of Done erstellen | für bessere Codequalität |

Tabelle 3 Verbesserungsliste 1



Abbildung 4 Sailboat Retrospektive

3 Sprint 2

3.1 Sprint Planning am 08.10.2020

Erster Punkt des Meetings war die Priorisierung der User Stories für den kommenden Sprint durch den Product Owner.

Aus den priorisierten Stories konnte das Team dann verschiedene Tasks für den Sprint ableiten (siehe Sprint Backlog 2).

Zur Schätzung der Tasks wurde erneut auf die Methode des Planning Pokers zurückgegriffen, allerdings diesmal mit dem Kartensatz 2. Bei Unstimmigkeiten über die Gewichtung eines Tasks, wurde offen im Team über die Unterschiede diskutiert um eine Einigung zu erzielen.

Zuletzt wurden dann noch die Tasks an die einzelnen Teammitglieder verteilt. Dabei konnte sich jedes Teammitglied abhängig von seinem Können, einen oder mehrere Tasks auswählen.

Sprint Backlog 2

Priorisierung User Story

- 1 Als Spieler möchte ich bequem von Zuhause aus Poker spielen können, damit ich mich nicht mit Corona anstecke.
- 2 Als Spieler möchte ich sowohl gegen reale Player als auch Bots spielen können, um das Spiel abwechslungsreich zu gestalten und um auch mit Freunden/alleine spielen zu können.
- 3 Als Spieler möchte ich die Höhe der Blinds und das Tischmaximum festlegen können, um das Spiel an meine Bedürfnisse anpassen zu können.
- 4 Als Spieler möchte ich das Startguthaben festlegen können, um länger spielen zu können und das Spiel interessanter zu gestalten.

Task	Aufwand	Verantwortlich
Gamecontroller schreiben	32	Luca
Dealer-Klasse		
Ausgabe-Klasse		

Human Player erstellen	16	Noah
Rules schreiben	32	Felix
Funktion um Hand zu bewerten		
Funktion um Hände zu vergleichen		
Menü	8	Sophia
Tischmaximum		
Startguthaben		
Tests schreiben	16	Miriam
UML-Diagramm		Team
Dokumentation		Miriam + Sophia

Tabelle 4 Sprint Backlog 2

3.2 Sprint Review am 22.10.2020

Zu Beginn des Meetings konnte diesmal ein lauffähiges Programm vorgestellt werden. Daraus abgeleitet wurden verschiedene Verbesserungen, die noch am Programm gemacht werden sollten.

Diese wurden in der Sprint Backlog Review Liste 2 zusammengefasst.

Sprint Backlog Review Liste 2

- 1 UML-Diagramm: An aktuelles Programm anpassen
- 2 Ausgabe: Sonderzeichenausgabe an ASCII-Zeichensatz anpassen
- 3 Tests: ausbauen
- 4 Rules: fertigstellen um sie einbinden zu können
- 5 Menü: Verbindung zum GameController herstellen

Tabelle 5 Sprint Backlog Review Liste 2

3.3 Sprint Retrospektive am 22.10.2020

Zur Durchführung dieser Review wurde eine Hero-Retrospektive ausgewählt, in der die verschiedenen Schlagworte zum abgelaufenen Sprint gesammelt werden konnte. Daraus abgeleitet wurde dann eine Verbesserungsliste 2, die Punkte enthält, welche dann in den nächsten Sprints oder auch in zukünftigen Projekten umgesetzt werden sollten.

Verbesserungsliste 2

- | | |
|--|--|
| 1 häufiger auf Pairprogramming zurückgreifen | v.a. zu Projektbeginn und bei komplexen Funktionen |
| 2 UML-Diagramm in neuen Projekten zu Beginn gemeinsam erstellen | einzelne Bausteine werden besser aufeinander abgestimmt und verbessert
Testbarkeit der Software |
| 3 im Sprint früher mit der Fertigstellung von Aufgaben beginnen | |
| 4 Abhängigkeit von Aufgaben von anderen Teammitgliedern reduzieren | |

Tabelle 6 Verbesserungsliste 2

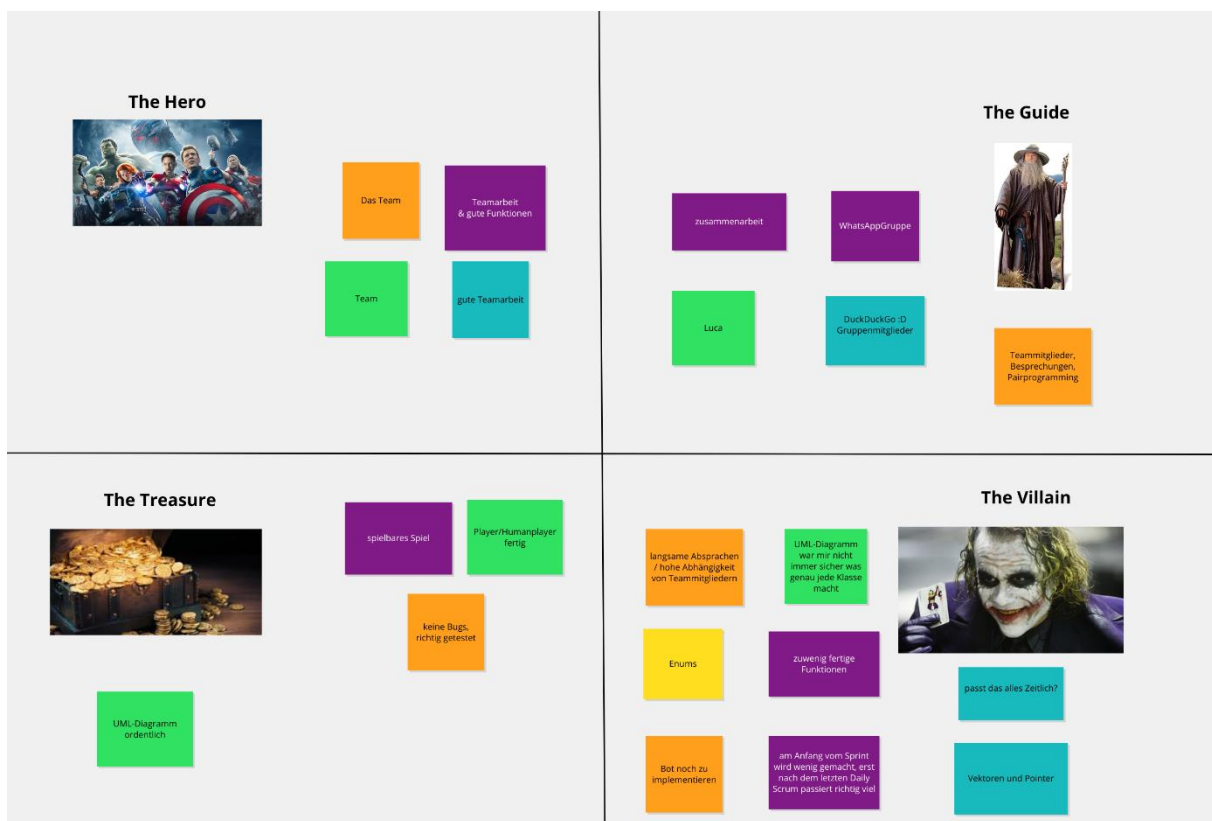


Abbildung 5 Helden Retrospektive

4 Sprint 3

4.1 Sprint Planning am 22.10.2020

Zu Beginn des Meetings musste die Rolle des Product Owners neu vergeben werden, da der vorherige Product Owner exmatrikuliert wurde. Als neuer Product Owner wurde Sophia Schmoll ernannt. Danach wurden neue User Stories in den Product Backlog eingebracht und durch den Product Owner priorisiert. Daraus wurden vom Team verschiedene Tasks abgeleitet (siehe Sprint Backlog 3), die dann mit Planning Poker bewertet wurden. Erneut wurde der Kartensatz 2 gewählt. Dieser hat sich im letzten Sprint als gut und genau erwiesen. In den letzten Sprint Plannings hat es sich bewährt, bei Unstimmigkeiten nochmals den genauen Umfang des Tasks zu besprechen, um dadurch eine Einigung zu erzielen.

Abschließend wurden die Tasks noch auf die einzelnen Teammitglieder verteilt. In dieser späten Phase des Projektes wurde vor allem bei der Verteilung darauf geachtet, dass die neuen Tasks an die Teammitglieder verteilt wurden, die in den vergangenen Sprints schon in diesem Bereich gearbeitet haben.

Sprint Backlog 3

Priorisierung User Story

- 1 Als Entwickler möchte das Menü und den Gamecontroller miteinander verknüpfen, um das Spiel spielbar zu machen
- 2 Als Spieler möchte ich einen schlaunen und dummen Bot zur Auswahl haben, um mein Spiel aufregend zu gestalten
- 3 Als Spieler möchte ich sowohl gegen reale Player als auch Bots spielen können, um das Spiel abwechslungsreich zu gestalten und um auch mit Freunden/alleine spielen zu können.
- 4 Als Entwickler möchte ich das Startguthaben der Bots zufällig festlegen, um ein aufregendes Spiel zu haben
- 5 Als Entwickler möchte ich, dass die anderen Entwickler Code kommentieren, um ihn besser verstehen zu können
- 6 Als Spieler möchte ich die Höhe der Blinds und das Tischmaximum festlegen können, um das Spiel an meine Bedürfnisse anpassen zu können.

- 7 Als Spieler möchte ich die Eingabemöglichkeiten über die Konsole auswählen können, damit das Spiel weiß was ich möchte.
- 8 Als Entwickler möchte ich die Ausgabe in englischer Sprache gestalten, um eine sprachliche Einheit darzustellen und die Ausgabe ordentlich zu gestalten.
- 9 Als Spieler möchte ich bequem von Zuhause aus Poker spielen können damit ich mich nicht mit Corona anstecke.

Task	Aufwand	Verantwortlich
Gamecontroller schreiben	8	Luca
Gewinnausgabe		
Regeln abfragen		
Dummen Bot erstellen	8	Luca
Schlauen Bot erstellen	64	Noah + Luca
Rules schreiben	32	Felix
Menü	16	Sophia
Verbindung zum Gamecontroller		
dummen Bot		
erstellen		
Tests schreiben	16	Miriam
Kommentare schreiben		Team
UML-Diagramm		Team
Dokumentation		Miriam + Sophia

Tabelle 7 Sprint Backlog 3

4.2 Sprint Review am 05.11.2020

Eingeleitet wurde das Meeting mit einer Demonstration der lauffähigen Software. Dabei konnten noch einige Punkte festgehalten werden, die im kommenden Sprint verbessert werden sollten. Diese wurden in festgehalten in der Sprint Backlog Review Liste 3.

Sprint Backlog Review Liste 3

- 1 Menu: 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)

Tabelle 8 Sprint Backlog Liste 3

4.3 Sprint Retrospektive am 05.11.2020

Die dritte Sprint Retrospektive wurde im Stil einer Amazon Produktbewertung durchgeführt. Dazu war jedes Teammitglied aufgefordert, eine Rezension zum Sprint zu schreiben und diesen mit einer Menge aus 5 Sternen zu bewerten. Die abgeleiteten Ergebnisse aus der Retrospektive wurden in der Verbesserungsliste 3 festgehalten.

Verbesserungsliste 3

- 1 Tasks zum Sprintende auch wirklich fertigstellen
- 2 Task am Anfang des Sprints beginnen, nicht erst zum Ende des Sprints hin
- 3 Offener Reden und dann direkt erkennen, dass ein Task zu kompliziert wird für den Sprint
- 4 Branches aktuell halten
- 5 Nur kommentierten und kompilierten Code pushen

Tabelle 9 Verbesserungsliste 3



Rezension:

- + wurde mehr über den Sprint verteilt gearbeitet (also nen bisschen)
- Nicht jeder nimmt Sprintende so genau(Rules, SmartBot)
- + Teamarbeit
- + zu Daily Scrums waren alle halbwegs Pünktlich



Rezension:

Alle waren Pünktlich zu den DailyScrums und wir sollten fast fertig geworden sein.



Rezension:

- +Die Arbeit wurde gut aufgeteilt,
- jedoch war der smarter Bot nicht machbar und wär fast ein Projekt für sich.
- +Alle waren Pünktlich zu den DailyScrums.
- Arbeit wurde am Ende des Sprints gemacht (wäre es früher gemacht worden, wäre das mit dem Bot auch früher erkannt worden)



Rezension:

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.



Rezension:

Ergebnis ist ein gutes Produkt, es gibt noch einige Bugs, die man verbessern könnte. Besonders schön finde ich, dass das Team gut zusammenarbeitet. Verbessert werden könnten noch die Aktualität der Mitgliederbranches um Mergefehler zu vermeiden. Außerdem sollten nur Branches gemergt werden, die auch wirklich compilieren und kommentiert wurden. Dennoch ein sehr gelungenes wenn auch noch nicht komplett fertiges Produkt.

Abbildung 6 Amazon Produktbewertung Retrospektive

5 Sprint 4

5.1 Sprint Planning 05.11.2020

Da in der letzten Sprint-Review nur wenige Verbesserungen gefunden wurden, die noch umgesetzt werden sollten, entschied sich das Team dazu, den Sprint 4 kürzer zu fassen als die restlichen Sprints. Diese Entscheidung ist auch darauf zurückzuführen, dass ein Großteil der User Stories bereits abgearbeitet war. Die verbleibenden UserStories wurden durch den Product Owner ein letztes Mal priorisiert. Die Tasks konnten in diesem Fall auch nicht wie in den vorherigen Planning vollständig aus den User Stories abgeleitet werden. Zwei große Tasks, für die das gesamte Team zuständig war, waren Bug Fixing und Kommentare schreiben. Da eine Aufwandsschätzung von Tasks für das gesamte Team, aus dem jedes Teammitglied nur bestimmte Teilaufgaben entnimmt, wenig sinnvoll ist, wurde in diesem Meeting auf ein Planning Poker verzichtet. Die Ergebnisse des Meetings sind im Sprint Backlog 4 zusammengefasst.

Sprint Backlog 4

Priorisierung User Story

- 1 Als Spieler möchte ich bequem von Zuhause aus Poker spielen können damit ich mich nicht mit Corona anstecke.
- 2 Als Entwickler möchte ich, dass die anderen Entwickler Code kommentieren, um ihn besser verstehen zu können
- 3 Als Spieler möchte ich einen schlaunen und dummen Bot zur Auswahl haben, um mein Spiel aufregend zu gestalten.
- 4 Als Spieler möchte ich sowohl gegen reale Player als auch Bots spielen können, um das Spiel abwechslungsreich zu gestalten und um auch mit Freunden/alleine spielen zu können.

Task	Verantwortlich
BugFixing	Team

Tests schreiben	Miriam
Kommentare schreiben	Team
UML-Diagramm	Team
Dokumentation	Miriam + Sophia
(Schlaufen Bot erstellen)	Noah + Luca

Tabelle 10 Sprint Backlog 4

5.2 Sprint-Review

Nach Abschluss des Sprints fand auch eine abschließende Produktreview statt. Dazu wurde die lauffähige Software präsentiert. Auch in dieser Review sind noch einige Verbesserungen am Programm aufgefallen. Diese wurden in der Product Backlog Review Liste 4 gesammelt. Diese könnten im Folgenden dazu verwendet werden, um über eine Verlängerung des Projektes zu beraten, oder könnten mit dem fertigen Produkt an den Kunden übergeben werden.

Sprint Backlog Review Liste 4

- 1 GameController: beim All-In wird nicht die richtige Summe zum Pot hinzugefügt
- 2 GameController: Unentschieden wird noch nicht in allen Fällen richtig entschieden

Tabelle 11 Sprint Backlog Review Liste 4

5.3 Sprint-Retrospektive

Als abschließende Retrospektive wurde vom Scrum Master ein Happiness Histogramm ausgewählt. Dabei konnte jedes Mitglied seinen Sticker an dem Platz auf der Achse platzieren, der seiner Stimmung entspricht. Ebenfalls bestand die Möglichkeit seine Platzierung des Stickers noch mit einem weiteren Sticker zu begründen. Daraus abgeleitete Verbesserungen, die dann im nächsten Projekt umgesetzt werden können, wurden in der Verbesserungsliste 4 festgehalten.

Verbesserungsliste 4

- 1 Mehr Review-Prozesse schon ins Daily-Scrum einbringen
- 2 Pull Requests genauer ansehen

Tabelle 12 Verbesserungsliste 4



Abbildung 7 Happiness Histogramm Retrospektive

6 UML Diagramm

Wie aus den Sprint Plannings hervorgeht, sollte das UML Diagramm eigentlich im Sprint 1 erstellt werden. In der Retrospektive wurde jedoch klar, dass es zum damaligen Stand nicht möglich war, das UML Diagramm fertig zu stellen. Es fehlten noch zu viele Verbindungen und einige Implementierungsschritte waren noch nicht klar.

Das Team einigte sich darauf, das UML Diagramm im Laufe des Projektes mit zu entwickeln. Dies wurde in Teilen auch umgesetzt. Nach einem Software-Problem wurde das UML Diagramm im letzten Sprint erneut begonnen und dann auch fertig gestellt. Das vollständige UML Diagramm ist in Abbildung 7 gezeigt. Zusätzlich ist es in GitHub abgelegt da somit eine Qualitativ hochwertigere Abgabe gewährleistet werden kann.

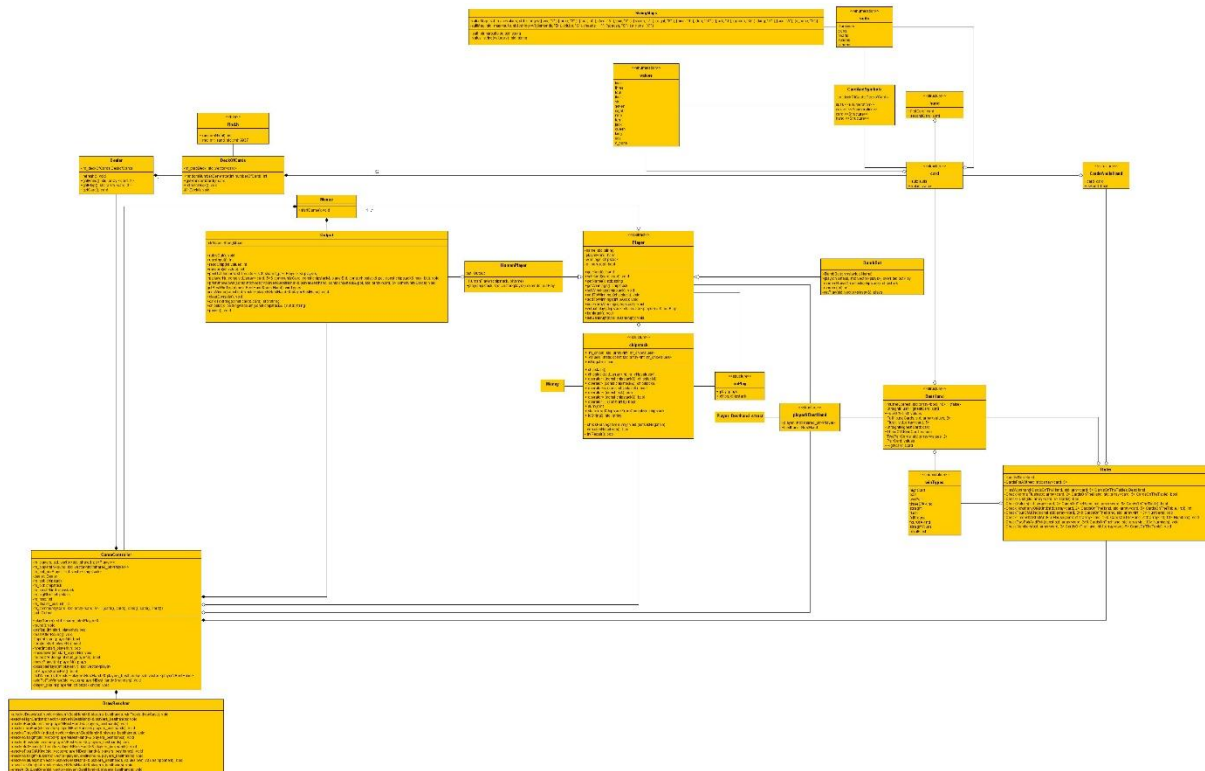


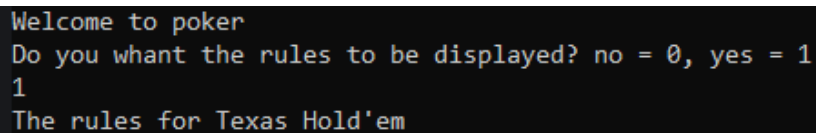
Abbildung 8 UML Diagramm

7 Ergebnis

7.1 Programm

Im Folgenden wird das fertige Produkt vorgestellt. Fertig ist dabei auf unsere User Stories von Beginn des Projektes zu beziehen, welche bis zum Ende des letzten Sprints Großteils umgesetzt werden konnten. Es ist natürlich noch einiger Spielraum für Verbesserungen gegeben, diese waren aber für die Dauer des Projektes nicht vorgesehen. Eine mögliche Erweiterung wäre ein Graphical User Interface. In User Story 3 wurde deshalb definiert, dass die Eingabe über die Konsole geregelt werden soll.

Zu Spielbeginn hat der Nutzer die Möglichkeit auszuwählen, ob er die Regeln angezeigt haben möchte (siehe User Story 2).



```
Welcome to poker
Do you want the rules to be displayed? no = 0, yes = 1
1
The rules for Texas Hold'em
-----
```

Abbildung 9 Spielbeginn

Anschließend kann die Zahl der Spieler, das Tischmaximum und die Art der Spieler festgelegt werden. Dies setzt die User Stories 5, 7 und 8 um. Die Art der Spieler bezieht sich dabei auf einen menschlichen Spieler oder einen Bot. Für menschliche Spieler muss ein Name und ein Chipstack eingegeben werden, dies wurde in User Story 13 beschrieben. Für einen Bot kann optional ein Name eingegeben werden. Das Startguthaben der Bots wird automatisch festgelegt (siehe User Story 9). Wie das in der Anwendung realisiert wurde, ist in Abbildung 9 zu sehen.

```
Please enter the number of players:
The maximum number of players is 9
3
Please enter the table range:
The minimal table range is 25000
280000
Should player 1 be a bot or a human?
0 = human; 1 = bot
0
Please enter a name for player 1:
Rubeus
Chipvalue: 5
amount: 6
Chipvalue: 10
amount: 5
Chipvalue: 20
amount: 4
Chipvalue: 50
amount: 3
Chipvalue: 100
amount: 2
Chipvalue: 500
amount: 1
Should player 2 be a bot or a human?
0 = human; 1 = bot
1
Du you whant to name the bot? 0 = yes, 1 = no
0
Please enter a name for player 2:
Silvanus
Should player 3 be a bot or a human?
0 = human; 1 = bot
1
Du you whant to name the bot? 0 = yes, 1 = no
1
Bigblind: { 5$: 0, 10$: 1, 20$: 0, 50$: 0, 100$: 0, 500$: 0 }
Smallblind: { 5$: 1, 10$: 0, 20$: 0, 50$: 0, 100$: 0, 500$: 0 }
Do you want to change the blinds? 0 = no, 1 = yes
0
```

Abbildung 10 Spielinitialisierung

Anschließend wird das eigentliche Spiel gestartet, die Bots spielen dabei automatisch und jeder menschliche Spieler hat die Möglichkeit über die Eingabe in der Konsole seine Aktionen auszuwählen. Somit wurde auch die Verknüpfung von Menü und GameController wie in User Story 12 beschrieben wurde realisiert. Das ist in Abbildung 10 zu sehen.

```
Your turn player Rubeus

hand:          { [S: 4] [C: Q] }
community cards: { [X: X] [X: X] [X: X] [X: X] [X: X] }

your Bid:      { 5$: 0, 10$: 0, 20$: 0, 50$: 0, 100$: 0, 500$: 0 } (sum: 0$)
highest Bid:   { 5$: 0, 10$: 1, 20$: 0, 50$: 0, 100$: 0, 500$: 0 } (sum: 10$)
Pot:           { 5$: 1, 10$: 1, 20$: 0, 50$: 0, 100$: 0, 500$: 0 } (sum: 15$)
your Money:    { 5$: 6, 10$: 5, 20$: 4, 50$: 3, 100$: 2, 500$: 1 } (sum: 1010$)

other Players:
  Silvanus:    { 5$: 7, 10$: 17, 20$: 17, 50$: 7, 100$: 1, 500$: 17 } (sum: 9495$)
  Bot1:        { 5$: 9, 10$: 9, 20$: 2, 50$: 10, 100$: 12, 500$: 12 } (sum: 7875$)
What do you want to do fold [0] check [1] raise [2]_
```

Abbildung 11 GameController

Die Spielmöglichkeiten für den Spieler sind ebenfalls in Abbildung 10 dargestellt. Um seinen Spielzug bestmöglich zu planen werden dem Spieler wie in Abbildung 10 dargestellt verschiedene Informationen über das Spiel aufgezeigt. In Abbildung 11 wurde beispielhaft erhöht.

```
What do you want to do fold [0] check [1] raise [2]2
How much do u want to raise?Chipvalue: 5
amount: 2
Chipvalue: 10
amount: 0
Chipvalue: 20
amount: 0
Chipvalue: 50
amount: 0
Chipvalue: 100
amount: 0
Chipvalue: 500
amount: 0
```

Abbildung 12 Erhöhung des Wetteinsatzes

Das Spielende ist durch einen Showdown (siehe Abbildung 12) dargestellt.

Anschließend wird der Gewinner der Runde angezeigt (siehe Abbildung 13)

```
SHOWDOWN:

community cards: {[C: 7] [C: 2] [D: 3] [C: K] [D: Q] }

pot: 3060

Rubeus: pair
      hand: { [S: 4] [C: Q] }
Silvanus: high Card
      hand: { [S: J] [C: 10] }
Bot1: pair
      hand: { [H: 10] [D: K] }
Press any key to continue . . .
```

Abbildung 13 Showdown

```
WINNERS:

      Rubeus
Press any key to continue . . .
```

Abbildung 14 Winners

Wie aus dem Abgleich des eben vorgestellten Ergebnisses mit der anfänglichen Beschreibung durch die User Stories hervorgeht, wurden alle User Stories umgesetzt. Bis auf User Story 10, da kein schlauer Bot implementiert wurde. Diese Aufgabe war im vorletzten Sprint gestellt und nach Überlegen von Implementierungsmöglichkeiten fallengelassen, da dem Bug Fixing eine höhere Priorität zugeordnet wurde und somit die Zeit fehlte diesen Bot noch zu implementieren. Einen schlauen Bot zu implementieren wäre genau wie ein GUI eine mögliche Erweiterung des Projektes.

7.2 Tests

Zur Eliminierung von Fehlerquellen wurden für das Projekt über 80 Tests geschrieben. Die Tests sollten vor allem Fehler in leicht zu testenden Funktionen verhindern. Zur Implementierung der Tests wurden das Google Mock und Google Test Framework verwendet.

Um einen besseren Überblick über die Tests zu haben, wurden die Tests nach Klassen und Funktionalität geordnet.

Wie in Abbildung 15 zu sehen ist, werden alle Tests erfolgreich ausgeführt.

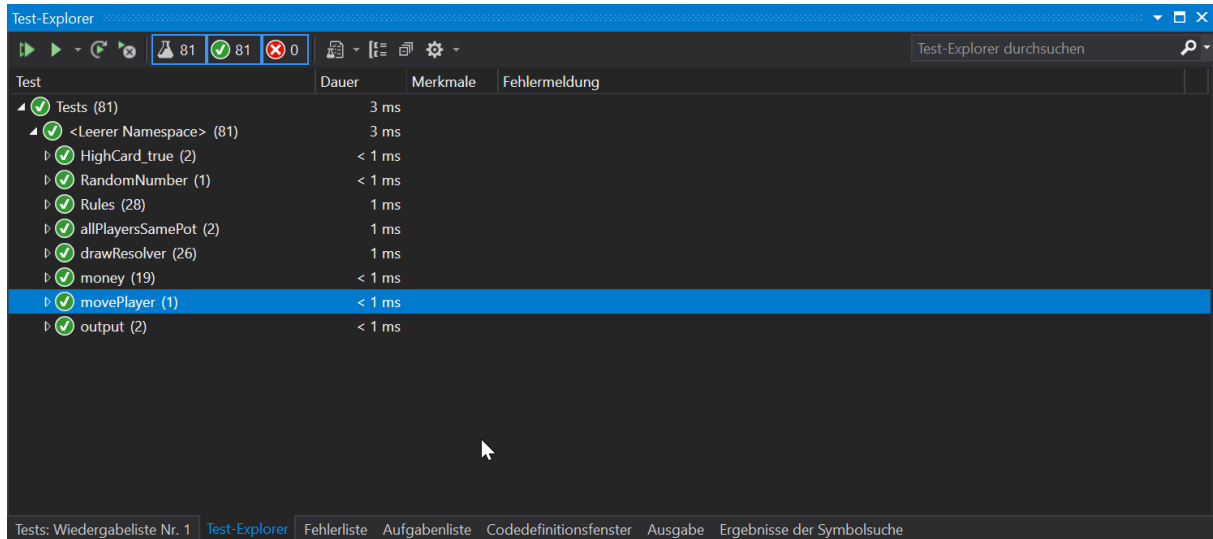


Abbildung 15 Tests

8 Zusammenfassung

Zusammenfassend lässt sich sagen, dass das Projekt eine gute Plattform bot, sich mit verschiedenen Punkten der Softwareentwicklung auseinanderzusetzen.

Zum einen konnten Erfahrungen in dem Scrum-Modell gesammelt werden. Welches dann gegen Projektende immer gewohnter für die Teammitglieder wurde und somit auch immer besser funktioniert hat.

Desweiteren musste zu Beginn eine Plattform gefunden werden, um zusammen an einer Softwareanwendung zu arbeiten. Durch die Wahl von GitHub, arbeiteten manche Teammitglieder erstmalig mit GitHub und andere konnten ihre Erfahrungen vertiefen.

Zudem konnten wir unsere Programmierkenntnisse erweitern. Das Projekt bot für den Teil der Gruppe, welcher im zweiten Semester Java gelernt hatte, die Möglichkeit sich mit C++ zu beschäftigen. Der Teil der Gruppe, welcher im zweiten Semester die Vorlesung C++ belegt hatte, konnte durch das Projekt die Vorhandenen Kenntnisse und Fertigkeiten erweitern und weitergeben.