

ID	Use-Case	Test-Name	Beschreibung	Beteiligte Klassen	Art	Status	letzter Testzeitpunkt
T001	UC-Game-01	testStartLineUp	Nachdem das Spiel durch den Host gestartet wurde, hat jeder Spieler drei Figuren im Wartebereich, eine Figur auf dem Startfeld und eine zufällige Sonderkarte auf der Hand.	Game, Board, Player, BonusCard, BoardView, GUIView	JUnit	erfolgreich	
T002	UC-Game-01	testCreatePowerCardDeck	Ein gemischtes Deck mit einer bestimmten Anzahl jeder Bonuskarte wird erstellt.	Game, BonusCard, BoardView, GUIView	JUnit	erfolgreich	
T003	UC-Game-02	testGameFinishes	Alle außer ein Spieler haben ihre vier Figuren in das Haus gebracht. Das Spiel wird beendet und die Siegerehrung beginnt.	Game, Board, , Piece, Node, BoardView, GUIView	JUnit	erfolgreich	
T004	UC-Game-03	testPlayerFinishes	Ein Spieler hat alle Figuren ins Haus gebracht und ist somit fertig. Mit den Figuren kann nun nicht mehr interagiert (ziehen, schlagen, tauschen, etc.) werden und der Spieler wird in den folgenden Runden übersprungen.	Game, Board, , PlayerState, Piece, Node, BoardView, GUIView	JUnit	erfolgreich	
T005	UC-Game-04	testAllPiecesInWaitingArea	Alle verbleibenden Figuren des Spielers sind im Wartebereich. Es können keine Bonuskarten verwendet werden und der Spieler benötigt eine 6 um ziehen zu können.	Game, Board, , Piece, BoardView, GUIView	JUnit	erfolgreich	
T006	UC-Game-04	test3TriesFor6	Sind alle verbleibenden Figuren eines Spielers im Wartebereich, hat dieser 3 Versuche eine 6 zu würfeln.	Game, ServerGameLogic	JUnit	erfolgreich	
T007	UC-Game-05	testGameTerminates	Wird ein laufendes Spiel vom Host beendet, wird der Spielstand verworfen und die Spieler kehren zum Start Dialog zurück.	ServerGameLogic, ClientGameLogic, BoardView, GUIView	Händisch	erfolgreich	
T008	UC-Game-06	testStartingOrder	Wurde das Spiel geladen, dürfen alle Spieler einmal würfeln. Der Spieler mit der höchsten Augenzahl darf beginnen und die Zugreihenfolge verläuft von ihm aus im Uhrzeigersinn.	Game, ServerGameLogic, BoardView, GUIView	JUnit	erfolgreich	
T009	UC-Game-07	testDouble	Beim Auswürfeln der Reihenfolge haben zwei oder mehr Spieler die gleiche Augenzahl geworfen und die verbliebenen Spieler haben niedrigere Augenzahlen geworfen. Die Spieler mit der gleichen Augenzahl würfeln erneut.	Game, ServerGameLogic, BoardView, GUIView	JUnit	erfolgreich	
T010	UC-Game-08	testChangeActivePlayer	Hat der aktive Spieler seinen Zug beendet, wird der nächste Spieler in der Zugreihenfolge zum aktiven Spieler.	Game, ServerGameLogic, BoardView, GUIView	JUnit	erfolgreich	
T011	UC-Game-09	testUseTurbo	Wurde eine Turbokarte verwendet, beträgt die Zahl der zu ziehenden Felder der gewürfelten Augenzahl verrechnet mit dem erhaltenen Multiplikator.	Game, BoardView, GUIView, ServerGameLogic	Händisch	erfolgreich	
T012	UC-Game-09	testMuliplicationChance	Die Multiplikatoren der Turbokarte haben eine 60% Chance auf x2, eine 20% Chance auf x1 und eine chance von 20% auf x0.	ServerGameLogic, Game	JUnit	erfolgreich	
T013	UC-Game-10	testTurboOn6	Wird eine 6 durch eine Turbokarte multipliziert und es sind noch Figuren im Haus, wird der Effekt eines 6er Wurfs gültig und eine Figur muss, sofern möglich, aus dem Wartebereich auf das Startfeld gesetzt werden und der Multiplikator verfällt.	Game, Board, ServerGameLogic, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T014	UC-Game-11	testAwardCeremony	Die Spieler werden der Siegerrangfolge nach auf einem Siebertreppchen platziert.	Game, GUIView	Händisch	erfolgreich	
T015	UC-Game-11	testStatistics	Im Anschluss an die Siegerehrung wird ein Bildschirm mit Spielstatistiken angezeigt, der eigene Name und der höchste Wert jeder Kategorie wird hervorgehoben.	Game, GUIView	Händisch	erfolgreich	
T016	UC-Game-12	testRefillPowerCardDeck	Ist der Bonuskartenstapel leer, werden die Karten aus dem Ablagestapel gemischt und dem Bonuskartenstapel hinzugefügt.	Game, BonusCard, BoardView, GUIView	JUnit	erfolgreich	
T017	UC-Player-01	testRollNo6AndCanMove	Der Spieler hat eine Augenzahl zwischen 1 & 5 gewürfelt und mindestens eine Figur, die nicht am ziehen gehindert wird. Eine der freien Figuren wird gewählt, welche um die gewürfelte Augenzahl zieht und der nächste Spieler ist am Zug.	Game, Board, , Node, PieceGame, BoardView, GUIView	Händisch	erfolgreich	
T018	UC-Player-02	testRoll6AndCanMove	Der Spieler hat eine 6 gewürfelt, keine Figur im Wartebereich und mindestens eine Figur, die nicht am Ziehen gehindert wird. Der Spieler wählt eine der freien Figuren, welche 6 Felder weiter zieht und der Spieler darf erneut würfeln.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T019	UC-Player-02	testDiceChance	Alle Augenzahlen haben eine Wahrscheinlichkeit von 1/6 geworfen zu werden.	ServerGameLogic	Händisch	erfolgreich	
T020	UC-Player-03	testRollNo6AndCantMove	Der Spieler hat eine Augenzahl zwischen 1 & 5 gewürfelt, jedoch kann keine Figur einen validen Zug machen. Sein Zug endet und der nächste Spieler ist dran.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T021	UC-Player-04	testRoll6AndCantMove	Der Spieler hat eine 6 gewürfelt und keine Figur im Wartebereich, jedoch kann keine Figur einen validen Zug machen. Er kann nicht ziehen und darf erneut würfeln.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T022	UC-Player-05	testSelectOwnPieceBeforeRollingDice	Spieler klickt eine eigene Figur an, bevor er würfelt. Die Figur wird hervorgehoben.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T023	UC-Player-05	testSwitchPieceBeforeRollingDice	Wird eine andere Figur angeklickt, wird die Hervorhebung der ersten Figur aufgehoben und die Neue wird hervorgehoben.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T024	UC-Player-06	testSelectOwnPieceAfterRollingDice	Spieler klickt eine eigene Figur an, nachdem er gewürfelt hat. Die Figur und das erreichbare Feld wird hervorgehoben.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T025	UC-Player-06	testSwitchPieceAfterRollingDice	Wird eine andere Figur angeklickt, wird die Hervorhebung der ersten Figur aufgehoben und die Neue wird hervorgehoben.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T026	UC-Player-07	testSelectOpponentPiece	Spieler klickt eine gegnerische Figur an. Die Figur wird hervorgehoben.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T027	UC-Player-07	testSwitchOpponentPiece	Wird eine andere Figur angeklickt, wird die Hervorhebung der ersten Figur aufgehoben und die Neue wird hervorgehoben.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T028	UC-Player-08	testDeselectPiece	Der Spieler wählt die zuvor ausgewählte Figur ab. Die Figur wird nun nicht mehr hervorgehoben.	Game, Board, , Node, Piece, BoardView, GUIView	Händisch	erfolgreich	
T029	UC-Player-09	testSelectBonusCard	Der Spieler wählt die Bonuskarte die er ausspielen will.	GUIView, ClientGameLogic,	Händisch	erfolgreich	
T030	UC-Player-10	testDeselectBonusCard	Der Spieler wählt die Bonuskarte ab, die er zuvor ausgewählt hat	GUIView, ClientGameLogic,	Händisch	erfolgreich	
T031	UC-Player-11	testPlayTurboCard	Der Spieler spielt die Turbo-Karte	GUIView, ClientGameLogic, ServerGameLogic, Game, View, Player, Bonuskarte	Händisch	erfolgreich	

ID	Use-Case	Test-Name	Beschreibung	Beteiligte Klassen	Art	Status	letzter Testzeitpunkt
T032	UC-Player-12	testPlayShieldCard	Der Spieler spielt die Schild-Karte	GUIView, ClientGameLogic, ServerGameLogic, Game, View, Player, Bonuskarte, Board, , Piece, Node	Händisch	erfolgreich	
T033	UC-Player-13	testPlaySwapCard	Der Spieler spielt die Tausch-Karte	GUIView, ClientGameLogic, ServerGameLogic, Game, View, Player, Bonuskarte, Board, , Piece, Node	Händisch	erfolgreich	
T034	UC-Player-14	testConfirmChoice	Der Spieler bestätigt die Auswahl seiner gewählten Bonuskarte oder der gewählten Spielfigur.	GUIView, ClientGameLogic	Händisch	erfolgreich	
T035	UC-Piece-01	testMove	Eine gewählte Spielfigur bewegt sich um den gewürfelten Wert im Uhrzeigersinn um das Spielfeld.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T036	UC-Piece-02	testCantMove	Eine gewählte Spielfigur kann sich nicht bewegen, da ihr Weg oder das Ziel blockiert sind. Es muss mit einer anderen Figur gezogen werden.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T037	UC-Piece-03	testNoPossibleMove	Kann keine Figur eine valide Bewegung machen, ist der Zug des Spielers beendet.	Game, Board, Node, Piece, , View	JUnit		
T038	UC-Piece-03	testThrow	Eine eigene Figur zieht auf ein Feld, auf welchem eine gegnerische Spielfigur, welche kein Schild besitzt, steht. Die eigene Figur bleibt auf dem Feld stehen, die gegnerische Figur wird in ihren Wartebereich gesetzt.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T039	UC-Piece-04	testGetThrown	Eine gegnerische Figur zieht auf ein Feld welches durch eine eigene Spielfigur, ohne Schild, besetzt wird. Die gegnerische Figur bleibt auf dem Feld stehen, die eigene Figur wird in ihren Wartebereich gesetzt.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T040	UC-Piece-05	testLeaveWaitingArea	Der Spieler hat Figuren im Wartebereich und eine 6 gewürfelt. Eine Figur aus dem Wartebereich wird auf das Startfeld gestellt.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T041	UC-Piece-06	testMustLeaveStartingField	Eine eigene Figur steht auf dem Startfeld, während noch Figuren im Wartebereich sind. Es muss mit der Figur auf dem Startfeld gezogen werden, sofern möglich.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T042	UC-Piece-07	testDontHaveToLeaveStartingField	Der Wartebereich ist leer. Eigene Figuren die auf dem Start stehen, müssen nicht gezogen werden.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T043	UC-Piece-08	testCantLeaveStartingField	Eine Figur, welche auf dem eigenen Startfeld steht kann mit der gewürfelten Augenzahl nicht ziehen. Ein anderer Zug muss ausgeführt werden.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T044	UC-Piece-09	testReachBonusField	Eine Figur zieht auf ein Bonusfeld. Der Nachziehstapel hat Bonuskarten. Der Spieler erhält eine zufällige Bonuskarte.	Game, Board, Node, Piece, , BonusCard, Player, View	JUnit	erfolgreich	
T045	UC-Piece-09	testNoPowerCards	Eine Figur zieht auf ein Bonusfeld. Der Nachziehstapel und der Ablagestapel sind leer. Der Spieler erhält keine Bonuskarte	Game, Board, Node, Piece, , BonusCard, Player, View	JUnit	erfolgreich	
T046	UC-Piece-09	testShufflePile	Eine Figur zieht auf ein Bonusfeld. Der Nachziehstapel ist leer und der Ablagestapel hat Karten. Der Ablagestapel wird gemischt und auf den Nachziehstapel gelegt. Der Spieler erhält eine Bonuskarte	Game, Board, Node, Piece, , BonusCard, Player, View	JUnit	erfolgreich	
T047	UC-Piece-10	testEnterHouse	Eine aktive Figur kann mit der gewürfelten Augenzahl, ein Feld im eigenen Haus erreichen. Die Figur zieht vom Spielfeld ins Haus.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T048	UC-Piece-10	testOnlyEnterOwnHouse	Eine Spielfigur kann nur in das Haus der eigenen TSK und nicht in die Häuser anderer TSKs.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T049	UC-Piece-11	testActiveHomePiece	Eine Figur im Haus kann bis auf das letzte, nicht belegte Feld ziehen.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T050	UC-Piece-11	testCantJumpOverFigureInHouse	Im Haus können Figuren nicht übersprungen werden.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T051	UC-Piece-12	testActiveHomePieceBlocked	Eine Figur im Haus kann sich nur bewegen, wenn die geworfene Augenzahl maximal der Anzahl freier Felder vor der Figur entspricht.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T052	UC-Piece-13	testOnStartingFieldWithShield	Eine Figur mit Schild, zieht auf ein Startfeld. Das Schild der Figur wird unterdrückt.	Game, Board, Node, Piece, View	JUnit	erfolgreich	
T053	UC-Piece-14	testThrowFigureWithShield	Eine andere Figur versucht auf das Feld einer, durch ein Schild geschützten Figur zu ziehen. Es passiert nichts und beide Figuren verbleiben an ihrem Platz.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T054	UC-Piece-15	testUseSwap	Spielt ein Spieler die Tausch-Karte, wechseln zwei ausgewählte Figuren ihre Position. Schild-Effekte bleiben erhalten.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T055	UC-Piece-16	testUseShield	Spielt ein Spieler die Schild-Karte und wählt eine Figur, erhält diese eine Schild, welches vor dem Schmeißen schützt und allen Spielern angezeigt wird.	Game, Board, Node, Piece, , View, BonusCard	JUnit	erfolgreich	
T056	UC-Piece-17	testLoseShield	Ist eine Runde vergangen, verliert die Spielfigur ihren Schild.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T057	UC-Piece-18	testFinishedPiece	Befindet sich eine Figur auf ihrer finalen Position, kann mit ihr nicht mehr interagiert werden.	Game, Board, Node, Piece, , View	JUnit	erfolgreich	
T058	UC-Server-01	testServerStart	Ein Spielserver wird auf der angegebenen Adresse mit dem angegebenen Port erstellt.	Server, NetworkDialog	Händisch	erfolgreich	
T059	UC-Server-01	testServerDoesntStart	Es konnte kein Server erstellt werden. Der Host erhält eine Fehlermeldung und kehrt zum Netzwerk-Dialog zurück.	Server, NetworkDialog, Client	Händisch	erfolgreich	
T060	UC-Server-02	testAcceptRequest	Der Server akzeptiert eine gesendete Beitrittsanfrage und fügt den Spieler der Lobby hinzu.	Server, Client	Händisch	erfolgreich	
T061	UC-Server-03	testTerminateServer	Der Server wird vom Host heruntergefahren.	Server, Client	Händisch	erfolgreich	
T062	UC-Server-04	testServerEndsGame	Wurden die Bedingungen für ein Spielende erfüllt, beendet der Server die Partie und beginnt mit der Siegerehrung.	Server, Client, ServerGameLogic, ClientGameLogic	Händisch	erfolgreich	
T063	UC-Server-05	testDeclineConnection	Versucht ein Spieler nach Verbindungsabbruch dem laufenden Server wieder beizutreten, wird dies vom Server abgelehnt.	Server, Client	Händisch	erfolgreich	
T064	UC-Server-06	testDeclineRequest	Der Server ist voll. Weitere Beitrittsanfragen werden abgelehnt.	Sever, Client	Händisch	erfolgreich	

ID	Use-Case	Test-Name	Beschreibung	Beteiligte Klassen	Art	Status	letzter Testzeitpunkt
T065	UC-Client-01	testClientTerminatesConnection	Wird eine Verbindung eigenmächtig getrennt, wird der Spieler als inaktiv markiert und kann nicht mehr am Spiel teilnehmen.	Server, ServerGameLogic, Client	Händisch	erfolgreich	
T066	UC-Client-02	testClientConnects	Der Client baut eine Verbindung mit dem Server auf und sendet eine Beitrittsanfrage.	Server, Client	Händisch	erfolgreich	
T067	UC-Client-03	testClientCantConnect	Der Client kann keine Verbindung zum Server aufbauen. Er erhält eine Fehlermeldung und kehrt zum Netzwerk-Dialog zurück.	Client, Server	Händisch	erfolgreich	
T068	UC-Client-04	testClientReconnects	Ein Client hat die Verbindung verloren und sich in der angegebenen Zeit wieder mit dem Server verbunden. Das Spiel wird unverändert fortgesetzt.	Server, Client, ServerGameLogic, ClientGameLogic	Händisch	erfolgreich	
T069	UC-Client-04	testClientDoesntReconnect	Ein Client hat die Verbindung verloren und sich in der angegebenen Zeit nicht wieder mit dem Server verbunden oder das Spiel wurde vorher vom Host fortgesetzt. Das Spiel wird fortgesetzt und der Spieler als inaktiv markiert.	Server, ServerGameLogic, Client	Händisch	erfolgreich	
T070	UC-Client-05	testClientDisconnects	Ein Client hat die Verbindung verloren. Der Server speichert den aktuellen Spielstand, pausiert das Spiel und benachrichtigt die Spieler.	Server, ServerGameLogic, Client, ClientGameLogic	Händisch	erfolgreich	
T071	UC-StartDialog-01	testEnterName	Der Spieler gibt seinen Spielernamen ein.	ClientGameLogic, Game, Player, View	JUnit	erfolgreich	
T072	UC-StartDialog-02	testJoinServer	Der Spieler möchte einem bestehenden Server beitreten. Es wird vom Start-Dialog zum Client Netzwerk-Dialog gewechselt.	Server, ServerGameLogic, Client, ClientGameLogic, View,	JUnit	erfolgreich	
T073	UC-StartDialog-03	testHostServer	Der Spieler möchte seinen eigenen Spielservers hosten. Es wird vom Start-Dialog zum Host Netzwerk-Dialog gewechselt.	Server, ServerGameLogic, Client, ClientGameLogic, View,	JUnit	erfolgreich	
T074	UC-StartDialog-04	testExitGame	Die Anwendung wird beendet.	Client, View, ClientGameLogic	Händisch	erfolgreich	
T075	UC-NetworkDialogHost-01	testSpecifyPort	Der Host wählt den Port für den Spielservers.	Client, ClientGameLogic, View	Händisch	erfolgreich	
T076	UC-NetworkDialogHost-02	testCreateServer	Der Host erstellt einen Spielservers mit den gegebenen Parametern und tritt diesem im Anschluss bei.	Client, ClientGameLogic, View, Server, ServerGameLogic	Händisch	erfolgreich	
T077	UC-NetworkDialogHost-03	testCancelServer	Der Host bricht die Erstellung eines eigenen Spielservers ab und kehrt zum Start-Dialog zurück.	Client, ClientGameLogic	Händisch	erfolgreich	
T078	UC-NetworkDialogClient-01	testEnterIP	Der Spieler gibt die IP-Adresse des Spielservers ein, welchem er beitreten möchte.	NetworkDialog, Client, ClientGameLogic, View	JUnit	erfolgreich	
T079	UC-NetworkDialogClient-02	testEnterPort	Der Spieler gibt den Port für den Spielservers ein, welchem er beitreten möchte.	NetworkDialog, Client, ClientGameLogic, View	Händisch	erfolgreich	
T080	UC-NetworkDialogClient-03	testConnectToServer	Der Client sendet eine Beitrittsanfrage zum angegebenen Server und der Spieler tritt der Lobby bei.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T081	UC-NetworkDialogClient-03	testCantConnectToServer	Der Beitritt schlägt fehl. Eine Fehlermeldung wird angezeigt und der Spieler ist wieder im Netzwor-Dialog.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T082	UC-NetworkDialogClient-04	testCancelJoining	Der Spieler bricht den Beitrittsvorgang ab und kehrt zum Start-Dialog zurück.	Client, ClientGameLogic, View	JUnit	erfolgreich	
T083	UC-Lobby-01	testSelectTSK	Der Spieler wählt eine der verfügbaren TSKs aus.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T084	UC-Lobby-02	testDeselectTSK	Der Spieler hebt die Auswahl seiner gewählten TSK auf.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T085	UC-Lobby-03	testChangeTSK	Der Spieler ändert seine gewählte TSK zu einer anderen, noch verfügbaren TSK.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T086	UC-Lobby-04	testReady	Hat der Spieler hat eine TSK gewählt und den "Bereit"-Knopf gedrückt, wird sein Status wird auf "Bereit" gesetzt.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T087	UC-Lobby-05	testNotReady	Drückt ein, als "Bereit" markierter, Spieler erneut den "Bereit"-Knopf, wird sein Status wieder auf "Nicht Bereit" gesetzt.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T088	UC-Lobby-06	testLeaveLobby	Der Spieler verlässt die Lobby und kehrt zum Start-Dialog zurück.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T089	UC-Lobby-07	testStartGame	Sind alle Spieler bereit, kann der Host das Spiel starten.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	teilweise	
T090	UC-Lobby-08	testShowStatus	Der Spielerstatus wird allen Spielern angezeigt.	Client, ClientGameLogic, Server, ServerGameLogic, View	Händisch	erfolgreich	
T091	UC-Lobby-09	testShowNames	Die Spielernamen werden allen Spielern angezeigt.	Client, ClientGameLogic, Server, ServerGameLogic, View	Händisch	erfolgreich	
T092	UC-Lobby-10	testShowAvailableTSKs	Die noch verfügbaren TSKs werden allen Spielern ausgegraut anzeigt.	Client, ClientGameLogic, Server, ServerGameLogic, View	Händisch	erfolgreich	
T093	UC-Lobby-11	testShowAssignedTSKs	Die nicht verfügbaren TSKs werden für alle Spieler farbig makiert.	Client, ClientGameLogic, Server, ServerGameLogic, View	Händisch	erfolgreich	
T094	UC-Lobby-12	testServerAssignsTSK	Drückt ein Spieler den "Bereit"-Button, ohne vorher eine TSK gewählt zu haben, wird ihm vom Server eine freie TSK zugewiesen.	Client, ClientGameLogic, Server, ServerGameLogic, View	JUnit	erfolgreich	
T095	UC-Settings-01	testOpenSettingsMenu	Durch drücken des "Zahnrad"-Buttons oder der Esc-Taste, wird der Dialog für die Einstellungen geöffnet.	Client, ClientGameLogic	Händisch	erfolgreich	
T096	UC-Settings-02	testCloseSetingsMenu	Durch drücken des Zahnrad-Buttons oder der Esc-Taste, wird der Dialog für die Einstellungen geschlossen.	Client, ClientGameLogic	Händisch	erfolgreich	

ID	Use-Case	Test-Name	Beschreibung	Beteiligte Klassen	Art	Status	letzter Testzeitpunkt
T097	UC-Settings-03	testOpenAudioSettings	Durch drücken des "Audio"-Buttons wird der Audio-Dialog geöffnet.	Client, ClientGameLogic	Händisch	erfolgreich	
T098	UC-Settings-04	testOpenVideoSettings	Durch drücken des "Video"-Buttons wird der Video-Dialog geöffnet.	Client, ClientGameLogic	Händisch	erfolgreich	
T099	UC-Settings-05	testLeaveGame	Durch drücken des "Spiel verlassen"-Buttons verlässt der Spieler das aktive Spiel und kehrt zum Start-Dialog zurück.	Client, ClientGameLogic	Händisch	erfolgreich	
T100	UC-AudioSettings-01	testChangeMasterVolume	In den Audioeinstellungen kann die Gesamt-Lautstärke verändert werden.	Client, ClientGameLogic	Händisch	erfolgreich	
T101	UC-AudioSettings-02	testChangeMusicVolume	In den Audioeinstellungen kann die Musik-Lautstärke verändert werden.	Client, ClientGameLogic	Händisch	erfolgreich	
T102	UC-AudioSettings-03	testChangeSoundEffectVolume	In den Audioeinstellungen kann die Effekt-Lautstärke verändert werden.	Client, ClientGameLogic	Händisch	erfolgreich	
T103	UC-AudioSettings-04	testCloseAudioSettings	Der Dialog mit den Audio-Einstellungen wird geschlossen und der Einstellungs-Dialog wieder angezeigt.	Client, ClientGameLogic	Händisch	erfolgreich	
T104	UC-VideoSettings-01	testDisplay	In den Video-Einstellungen kann zwischen Fenstermodus und Vollbild gewechselt werden.	Client, ClientGameLogic	Händisch	erfolgreich	
T105	UC-VideoSettings-02	testResolution	Das Spiel wird in der gewählten Auflösung angezeigt.	Client, ClientGameLogic	Händisch	erfolgreich	
T106	UC-VideoSettings-03	testCloseVideoSettings	Der Dialog mit den Video-Einstellungen wird geschlossen und der Einstellungs-Dialog wieder angezeigt.	Client, ClientGameLogic	Händisch	erfolgreich	
T107	UC-ClientView-01	testDisplayPiecesInWaitingArea	Figuren in den Wartebereichen, werden allen Spielern grafisch angezeigt.	BoardView, GUIView, Client, AppGameSte	Händisch	erfolgreich	
T108	UC-ClientView-02	testDisplayPiecesInGame	Figuren auf dem Feld (ohne Wartebereich / Haus) werden allen Spielern grafisch angezeigt.	BoardView, GUIView, Client, AppGameSte	Händisch	erfolgreich	
T109	UC-ClientView-03	testDisplayPiecesInHouse	Figuren in den Häusern, werden allen Spielern grafisch angezeigt.	BoardView, GUIView, Client, AppGameSte	Händisch	erfolgreich	
T110	UC-ClientView-04	testDisplayActivePlayer	Allen Spielern wird der aktive Spieler durch ein Würfel-Symbol angezeigt.	BoardView, GUIView, Client, AppGameSte	Händisch	erfolgreich	
T111	UC-ClientSound-01	testBackgroundMusic	Im Hintergrund wird Musik gespielt.	Music, ClientGameLogic, Client	Händisch	erfolgreich	
T112	UC-ClientSound-02	testDiceSoundEffect	Beim Würfeln wird ein entsprechender Sound abgespielt.	GameSound, ClientGameLogic, Client	Händisch	erfolgreich	
T113	UC-ClientSound-02	testMoveSoundEffect	Beim Ziehen wird ein entsprechender Sound abgespielt.	GameSound, ClientGameLogic, Client	Händisch	erfolgreich	
T114	UC-ClientSound-02	testThrowSoundEffect	Beim Schlagen wird ein entsprechender Sound abgespielt.	GameSound, ClientGameLogic, Client	Händisch	erfolgreich	
T115	UC-Camera-01	testRotateLeftKeyboard	Durch drücken der "Q"-taste dreht sich die Kamera nach links um das Spielfeld.	Client, BoardView, GUIView	Händisch	erfolgreich	
T116	UC-Camera-01	testRotateLeftMouse	Durch drücken der rechten Maustate und einer Mausbewegung nach links, dreht sich die Kamera nach links um das Spielfeld.	Client, BoardView, GUIView	Händisch	erfolgreich	
T117	UC-Camera-02	testRotateRightKeyboard	Durch drücken der "E"-Taste dreht sich die Kamera nach rechts um das Spielfeld.	Client, BoardView, GUIView	Händisch	erfolgreich	
T118	UC-Camera-02	testRotateRightMouse	Durch drücken der rechten Maustate und einer Mausbewegung nach rechts, dreht sich die Kamera nach rechts um das Spielfeld.	Client, BoardView, GUIView	Händisch	erfolgreich	
T119	UC-Camera-03	testZoomIn	Durch bewegen des Mausekaders nach oben, wird die Spielansicht hineingezoomt.	Client, BoardView, GUIView	Händisch	erfolgreich	
T120	UC-Camera-04	testZoomOut	Durch bewegen des Mausekaders nach unten, wird die Spielansicht herausgezoomt.	Client, BoardView, GUIView	Händisch	erfolgreich	
T121	UC-Camera-05	testStrategicView	Durch das Herauszoomen in der normalen Ansicht über einen bestimmten Threshold, wird in die strategische Ansicht gewechselt.	Client, BoardView, GUIView	Händisch	erfolgreich	
T122	UC-Camera-06	testNormalView	Durch das Hineinzoomen in der strategischen Ansicht über einen bestimmten Threshold, wird in die normale Ansicht gewechselt.	Client, BoardView, GUIView	Händisch	erfolgreich	
T123	UC-ServerState-01	testInitialStateServerState	Der Server befindet sich nach dem Start im Zustand Lobby.	ServerState, Lobby, ServerGameLogic	JUnit	erfolgreich	
T124	UC-ServerState-02	testLobbyToDetermineStartPlayer	Bei Erhalt der Message {LobbyReady} von allen Spielern wechselt der Server vom Zustand Lobby in den Zustand DetermineStartPlayer.	ServerState, Lobby, Game, DetermineStartPlayer, ServerGameLogic	JUnit	erfolgreich	
T125	UC-ServerState-03	testStayInLobby	Der Server bleibt bis zum Erhalt der Message {LobbyReady} im Zustand Lobby.	ServerState, Lobby, Game, DetermineStartPlayer, ServerGameLogic	JUnit	erfolgreich	
T126	UC-ServerState-04	testServerGameSubStatesToInterrupt	Verliert ein Spieler die Verbindung wechselt der Server aus jedem unterzustand von Game in den Zustand Interrupt.	ServerState, Game, Interrupt, ServerGameLogic	JUnit	erfolgreich	
T127	UC-ServerState-05	testServerGameToCeremony	Sind die Siegesbedingungen erfüllt, wechselt der Server vom Zustand Game in den Zustand Ceremony.	ServerState, Game, Ceremony, ServerGameLogic	JUnit	erfolgreich	
T128	UC-ServerState-06	testInterruptToGameContinue	Bei Erhalt der Message {ForceStartGame} wechselt der Server vom Zustand Interrupt in den Zustand Game.	ServerState, Game, Interrupt, ServerGameLogic	JUnit	erfolgreich	
T129	UC-ServerState-07	testInterruptToGameReconnect	Bei Erhalt der Message {ContinueGame} wechselt der Server vom Zustand Interrupt in den Zustand Game.	ServerState, Game, Interrupt, ServerGameLogic	JUnit	nicht implementiert	
T130	UC-ServerState-08	testInterruptToGameTimer	Nach Ablauf des Timers wechselt der Server automatisch vom Zustand Interrupt in den Zustand Game.	ServerState, Game, Interrupt, ServerGameLogic	JUnit	nicht implementiert	
T131	UC-ServerState-09	testCeremonyToServerStateEndState	Nach dem Senden der Message{Ceremony} wechselt der Server in den Endzustand.	ServerState, Ceremony, ServerGameLogic	Händisch	erfolgreich	
T132	UC-ServerState-10	testDeterminStartPlayerToDetermineStartPlayer1	Nach Erhalt der Message {RequestDice} bleibt der Server im Zustand DetermineStartPlayer	ServerState, ServerGameLogic, Game, DetermineStartPlayer	JUnit	erfolgreich	
T133	UC-ServerState-11	testDeterminStartPlayerToDetermineStartPlayer2	Haben zwei oder mehr Spieler das höchste Würfelergbnis, bleibt der Server im Zustand DetermineStartPlayer	ServerState, ServerGameLogic, Game, DetermineStartPlayer	JUnit	erfolgreich	

ID	Use-Case	Test-Name	Beschreibung	Beteiligte Klassen	Art	Status	letzter Testzeitpunkt
T134	UC-ServerState-12	testDeterminStartPlayerToAnimation	Nachdem der Startspieler feststeht, wechselt der Server in den AnimationState zur Anzeige des ersten Spielers.	ServerState, ServerGameLogic, Game, DetermineStartPlayer	JUnit	erfolgreich	
T135	UC-ServerState-13	testAnimationToPowerCard	Nach Anzeigen des aktiven Spielers im AnimatioState, wechselt der Server in den Zustand PowerCard.	ServerState, ServerGameLogic, Game, Animation, Turn	JUnit	erfolgreich	
T136	UC-ServerState-14	testTurnToAnimation	Nach Beendigung des Zugs eines Spielers wechselt der Server kurz in den Zustand Animation um die Zuganimation darzustellen.	ServerState, ServerGameLogic, Game, Turn, Animation	JUnit	erfolgreich	
T137	UC-ServerState-15	testTurnToGameEndState	Wurden die Siegesbedingungen im Turn eines Spielers erfüllt, wechselt der Server in den EndState von Game.	ServerState, ServerGameLogic, Game, Turn, Ceremony	JUnit	erfolgreich	
T138	UC-ServerState-16	testStayInPowerCard	Der Server bleibt bis Erhalt der Message {RequestDice} oder {RequestPlayCard} im Zustand PowerCard.	ServerState, ServerGameLogic, Game, Turn, Powercard	JUnit	erfolgreich	
T139	UC-ServerState-17	testPowerCardToPlayPowerCard	Schickt der Client die Message {RequestPlayCard}, wechselt der Server in den Zustand PlayPowerCard.	ServerState, ServerGameLogic, Game, Turn, Powercard, PlayPowerCrad	JUnit	erfolgreich	
T140	UC-ServerState-18	testPlayPowerCardToRollDice	Schickt der Client die Message {RequestDice}, wechselt der Server in den Zustand RollDice.	ServerState, ServerGameLogic, Game, Turn, PlayPowerCard, FirstRoll, RollDice	JUnit	erfolgreich	
T141	UC-ServerState-19	testChoosePieceToMovePiece	Wurde eine Figur gewählt und eine {RequestMove} Message vom Client an den Server geschickt, wechselt dieser in den Zustand MovePiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, MovePiece	JUnit	erfolgreich	
T142	UC-ServerState-20	testMovePiecetoTurnEndState	Wurde die Figur bewegt, ist der Zug des Spielers vorbei und der Server wechselt in den EndState von Turn.	ServerState, ServerGameLogic, Game, Turn, MovePiece	JUnit	erfolgreich	
T143	UC-ServerState-21	testMovePieceToFirstRoll	Wurde eine 6 gewürfelt, wechselt der Server nach dem Bewegen einer Figur wieder in den Zustand FristRoll.	ServerState, ServerGameLogic, Game, DetermineStartPlayer, Turn, RollDice, FirstRoll, RollDice	JUnit	erfolgreich	
T144	UC-ServerState-22	testFirstRollToRollDiceEndState	Hat der Spieler Figuren im Infield, wechselt der Server nach dem Würfeln in den EndState von RollDice und damit in den Zustand NoPiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, MovePiece, FirstRoll, RollDice	JUnit	erfolgreich	
T145	UC-ServerState-23	testFirstRollToSecondRoll	Hat der Spieler keine Figuren im Infield und keine 6 gewürfelt, wechselt der Server in den Zustand SecondRoll.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, SecondRoll, FirstRoll, RollDice	JUnit	erfolgreich	
T146	UC-ServerState-24	testSecondRollToRollDiceEndState	Wurde eine 6 gewürfelt, wechselt der Server in den EndState von RollDice und damit in den Zustand NoPiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, MovePiece, SecondRoll, RollDice	JUnit	erfolgreich	
T147	UC-ServerState-25	testSecondRollToThirdRoll	Wurde keine 6 gewürfelt, wechselt der Server in den Zustand ThirdRoll.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, SecondRoll, ThirdRoll, RollDice	JUnit	erfolgreich	
T148	UC-ServerState-26	testThirdRollToRollDiceEndState	Wurde eine 6 gewürfelt, wechselt der Server in den EndState von RollDice und damit in den Zustand NoPiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, MovePiece, ThirdRoll	JUnit	erfolgreich	
T149	UC-ServerState-27	testThirdRollToTurnEndState	Wurde keine 6 gewürfelt, wechselt der Server in den EndState von Turn und der Zug des Spielers ist vorbei.	ServerState, ServerGameLogic, Game, Turn, ThirdRoll	JUnit	erfolgreich	
T150	UC-ServerState-28	testNoPieceToWaitingPiece	Erfüllt ein Spieler die Bedingungen eine Figur aus dem Wartebereich zu holen, wechselt der Server in den Zustand WaitingPiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, NoPiece, WaitingPiece	JUnit	erfolgreich	
T151	UC-ServerState-29	testNoPieceToNoTurn	Spieler kann trotz gespielter Turbokarte keine Figur bewegen. Der Server wechselt in den Zustand NoTurn.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, NoPiece, NoTurn	JUnit	erfolgreich	
T152	UC-ServerState-30	testNoTurnToTurnEndState	Sind keine validen Spielzüge möglich wechselt der Server in den EndState von Turn und beendet somit den Zug des Spielers.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, NoPiece, NoTurn	JUnit	erfolgreich	
T154	UC-ServerState-31	testStayInWaitingPiece	Der Server ist in WaitingPiece, bekommt Nachrichten und verlässt diesen nicht	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, NoPiece, NoTurn	JUnit	erfolgreich	
T155	UC-ServerState-32	testWaitingPieceToMovePiece	Wurde eine Figur aus dem Wartebereich zum ziehen ausgewählt, wechselt der Server in den Zustand MovePiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, WaitingPiece	JUnit	erfolgreich	
T155	UC-ServerState-33	testNoPieceToSelectPiece	Wird vom Spieler eine Figur ausgewählt, wechselt der Server in den Zustand ChoosePiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, NoPiece, NoTurn	JUnit	erfolgreich	
T156	UC-ServerState-34	testNoPieceToStartPiece	Steht eine Figur auf dem Startfeld wird diese automatisch ausgewählt und der Server v	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, NoPiece, StartPiece	JUnit	erfolgreich	
T158	UC-ServerState-35	testStayInSelectPiece	Versucht ein Spieler invalide Figuren auszuwählen, wird er darüber benachrichtigt und der Server bleibt im Zustand SelectPiece, bis eine valide Auswahl getroffen wurde.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, SelectPiece	JUnit	erfolgreich	
T158	UC-ServerState-36	testSelectPieceToMovePiece	Wurde eine valide Figur gewählt, wechselt der Server in den Zustand MovePiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, SelectPiece, MovePiece	JUnit	erfolgreich	
T159	UC-ServerState-37	testStayInStartPiece	Der Server ist in StartPiece, bekommt Nachrichten und verlässt diesen nicht	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, StartPiece	JUnit	erfolgreich	
T160	UC-ServerState-38	testStartPieceToMovePiece	Die Figur auf dem Startfeld wurde angewählt und muss bewegt werden. Der Sever wechselt in den Zustand MovePiece.	ServerState, ServerGameLogic, Game, Turn, ChoosePiece, StartPiece, MovePiece	JUnit	erfolgreich	
T161	UC-ClientState-01	testInitalStateClientState	Der Client befindet sich nach dem Start im Zustand Dialogs.	Client, ClientState, ClientGameLogic, Dialogs, Startdialog	JUnit	erfolgreich	
T162	UC-ClientState-02	testDialogsToGame	Bei Erhalt der Message {StartGame} wechselt der Client vom Zustand Dialogs in den Zustand Game.	Client, ClientState, ClientGameLogic, Dialogs, Game	JUnit	erfolgreich	
T163	UC-ClientState-03	testDialogsToClientStateEndState	Der Client wechselt beim Beenden der Anwendung in den Endzustand von ClientState.	Client, ClientState, ClientGameLogic, Dialogs	Händisch	erfolgreich	
T164	UC-ClientState-04	testClientGameToCeremony	Bei Erhalt der Message {Ceremony} wechselt der Client vom Zustand Game in den Zustand Ceremony.	Client, ClientState, ClientGameLogic, Dialogs, Ceremony	JUnit	erfolgreich	

ID	Use-Case	Test-Name	Beschreibung	Beteiligte Klassen	Art	Status	letzter Testzeitpunkt
T165	UC-ClientState-05	testClientGameSubStatesToInterrupt	Bei Erhalt der Message {PauseGame} wechselt der Client vom Zustand Game in den Zustand Interrupt.	Client, ClientState, ClientGameLogic, Game, Interrupt	JUnit	erfolgreich	
T166	UC-ClientState-06	testGameToDialogs	Der Client wechselt beim Verlassen des Spiels vom Zustand Game in den Zustand Dialog.	Client, ClientState, ClientGameLogic, Dialogs, Game	JUnit	erfolgreich	
T167	UC-ClientState-07	testStayInInterrupt	Bis zum Erhalt der Message {ResumeGame}, bleibt der Client im Zustand Interrupt.	Client, ClientState, ClientGameLogic, Interrupt	JUnit	erfolgreich	
T168	UC-ClientState-08	testClientInterruptToGame	Bei Erhalt der Message {ResumeGame} wechselt der Client vom Zustand Interrupt in den Zustand Game.	Client, ClientState, ClientGameLogic, Game, Interrupt	JUnit	erfolgreich	
T169	UC-ClientState-09	testInterruptToDialogs	Verlässt ein Spieler während eines Interrupts das Spiel, wechselt der Client vom Zustand Interrupt zum Zustand Dialogs.	Client, ClientState, ClientGameLogic, Dialogs, Interrupt	JUnit	erfolgreich	
T170	UC-ClientState-10	testCeremonyToDialogs	Nach Beendigung der Siegerehrung, wechselt der Client vom Zustand Ceremony in den Zustand Dialogs.	Client, ClientState, ClientGameLogic, Dialogs, Ceremony	JUnit	erfolgreich	
T171	UC-ClientState-11	testStartDialogToNetworkDialog1	Möchte der Spieler ein Spiel hosten, wechselt der Client als Host in den Zustand NetworkDialog.	Client, ClientState, ClientGameLogic, Dialogs, Startdialog, NetworkDialog	JUnit	erfolgreich	
T172	UC-ClientState-12	testStartDialogToNetworkDialog2	Möchte der Spieler einem Spiel beitreten, wechselt der Client als Player in den Zustand NetworkDialog.	Client, ClientState, ClientGameLogic, Dialogs, Startdialog, NetworkDialog	JUnit	erfolgreich	
T173	UC-ClientState-13	testStartDialogToDialogsEndState	Drückt ein Spieler im Start Dialog den Button "Spiel Verlassen", wechselt der Client in den Endzustand von Dialogs.	Client, ClientState, ClientGameLogic, Dialogs, StartDialog	Händisch	erfolgreich	
T174	UC-ClientState-14	testNetworkDialogToStartDialog	Drückt ein Spieler im Network Dialog den Button "Zurück", wechselt der Client vom Zustand NetworkDialog in den Zustand StartDialog.	Client, ClientState, ClientGameLogic, Dialogs, Startdialog, NetworkDialog	JUnit	erfolgreich	
T175	UC-ClientState-15	testNetworkDialogToNetworkDialog1	Wird falsche Serverdaten angegeben, oder der Server sendet eine {LobbyDeny} Message, bleibt der Client im Zustand NetworkDialog	Client, ClientState, ClientGameLogic, Dialogs, NetworkDialog	JUnit	erfolgreich	
T176	UC-ClientState-16	testNetworkDialogToNetworkDialog2	Nach eingabe korrekter Serverdaten, bleibt der Client bis zum Erhalt der Message {LobbyAccept} im Zustand NetworkDialog.	Client, ClientState, ClientGameLogic, Dialogs, NetworkDialog	Händisch	erfolgreich	
T177	UC-ClientState-17	testNetworkDialogToLobby	Bei Erhalt der Message {LobbyAccept} wechselt der Client vom Zustand NetworkDialog in den Zustand Lobby.	Client, ClientState, ClientGameLogic, Dialogs, NetworkDialog, Lobby	JUnit	erfolgreich	
T178	UC-ClientState-18	testLobbyToStartDialog	Verlässt ein Spieler die Lobby, wechselt der Client vom Zustand Lobby zum Zustand StartDialog	Client, ClientState, ClientGameLogic, Dialogs, Startdialog, Lobby	JUnit	erfolgreich	
T179	UC-ClientState-19	testStayInLobby	Die Clients updaten die View entsprechend der ServerMessages {UpdateTsk}& {UpdateReady}	Client, ClientState, ClientGameLogic, Dialogs, Lobby	JUnit	erfolgreich	
T180	UC-ClientState-20	testLobbyToRollRankingDice	Wurder das Spiel durch die ServerMessage {StartGame} gestartet, wechselt der Client in den Zustand RollRankingDice.	Client, ClientState, ClientGameLogic, Dialogs, Lobby, Game, DetermineStartPlayer, RollRankingDice	JUnit	erfolgreich	
T181	UC-ClientState-21	testDetermineStartPlayerToWait	Die nicht aktiven Clients wechseln nach dem Auswürfeln der Zugreihenfolge in den Zustand Wait.	Client, ClientState, ClientGameLogic, Game, DetermineStartPlayer, Wait	JUnit	erfolgreich	
T182	UC-ClientState-22	testWaitToAnimation	Zum Abspielen von Animationen wechselt der Client kurzzeitig in den AnimationState.	Client, ClientState, ClientGameLogic, Game, Wait, Animation	JUnit	erfolgreich	
T183	UC-ClientState-23	testWaitToTurn	Wird der Client zum aktiven Spieler, wechselt er vom Zustand Wait in den Zustand Turn.	Client, ClientState, ClientGameLogic, Game, Turn, Wait	JUnit	erfolgreich	
T184	UC-ClientState-24	testWaitToGameEndState	Wurden die Siegesbedingungen erfüllt, wechselt der Client aus dem Zustand Wait in den EndState von Game.	Client, ClientState, ClientGameLogic, Game, Turn, Wait	JUnit	erfolgreich	
T185	UC-ClientState-25	testTurnSubStatesToGameEndState	Aus jedem Unterzustand von Turn kann bei Erfüllung der Siegesbedingungen in den EndState von Game gewechselt werden.	Client, ClientState, ClientGameLogic, Game, Turn	JUnit	erfolgreich	
T186	UC-ClientState-26	testTurnSubStatesToWait	Aus jedem Unterzustand von Turn kann der Client in den Zustand Wait wechseln.	Client, ClientState, ClientGameLogic, Game, Turn, Wait	JUnit	erfolgreich	
T187	UC-ClientState-27	testTurnSubStatesToSpectator	Aus jedem Unterzustand von turn, kann der Client in den Zustand Spectator wechseln.	Client, ClientState, ClientGameLogic, Game, Turn, Spectator	JUnit	erfolgreich	
T188	UC-ClientState-28	testSpectatorToGameEndState	Clients im Zustand Spectator wechseln bei Erfüllung der Siegesbedingungen in den EndState von Game.	Client, ClientState, ClientGameLogic, Game, Spectator	JUnit	erfolgreich	
T189	UC-ClientState-29	testPowerCardSubStatesToPlayPowerCard	Aus jedem Unterzustand von PowerCards kann in den Zustand PlayPowerCard gewechselt werden.	Client, ClientState, ClientGameLogic, Game, ChoosePowerCard, PlayPowerCard, PowerCard, Shield, Swap	JUnit	erfolgreich	
T190	UC-ClientState-30	testPowerCardSubStatesToRollDice	Soll keine PowerCard gespielt werden, kann aus jedem Unterzustand von PowerCard in den Zustand RollDice gewechselt werden.	Client, ClientState, ClientGameLogic, Game, ChoosePowerCard, PlayPowerCard, RollDice, PowerCard, Shield, Swap	JUnit	erfolgreich	
T191	UC-ClientState-31	testStayInPlayPowerCard	Der Client ist in PlayPowerCard, erhält Nachrichten und verlässt diesen nicht	Client, ClientState, ClientGameLogic, Game, PlayPowerCard	JUnit	erfolgreich	
T192	UC-ClientState-32	testPlayPowerCardToRollDice	Nach verwendung einer PowerCard wechselt der Client in den Zustand RollDice.	Client, ClientState, ClientGameLogic, Game, PlayPowerCard, RollDice	JUnit	erfolgreich	
T193	UC-ClientState-33	testStayInRollDice	Der Client bleibt für einen zweiten oder dritten würfelwurf im Zustand RollDice.	Client, ClientState, ClientGameLogic, Game, Turn, RollDice	JUnit	erfolgreich	
T194	UC-ClientState-34	testRollDiceToChoosePiece	Wurde ein valides Würfelergebnis geworfen, wechselt der Client in den Zustand ChoosePiece.	Client, ClientState, ClientGameLogic, Game, Turn, RollDice, ChoosePiece	JUnit	erfolgreich	
T195	UC-ClientState-35	testRollDiceToWait	Wurde kein valides Würfelergebnis geworfen, wechselt der Client in den Zustand Wait.	Client, ClientState, ClientGameLogic, Game, Turn, RollDice, Wait	JUnit	erfolgreich	
T196	UC-ClientState-36	testChoosePieceToWait	Der Client bekommt die Nachricht {NoTurn} in ChoosePiece und wechselt zu Turn	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, Wait	JUnit	erfolgreich	

ID	Use-Case	Test-Name	Beschreibung	Beteiligte Klassen	Art	Status	letzter Testzeitpunkt
T197	UC-ClientState-37	testChoosePieceToMovePiece	Der Client erhält die Message {MoveMessage} und wechselt in den MovePiece-State	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, MovePiece	JUnit	erfolgreich	
T198	UC-ClientState-38	testMovePieceToWait	Nach dem Bewegen der Figur ist der Zug des Clients zu Ende und er wechselt in den Zustand Wait.	Client, ClientState, ClientGameLogic, Game, Turn, MovePiece, Wait	JUnit	erfolgreich	
T199	UC-ClientState-39	testMovePieceToSpectator	Sind alle Figuren des Clients im Haus, erhält er die Message {Spectator} und wechselt er in den Zustand Spectator.	Client, ClientState, ClientGameLogic, Game, Turn, MovePiece, Spectator	JUnit	erfolgreich	
T200	UC-ClientState-40	testMovePieceToCeremony	Erhält der Client die ServerMessage {Ceremony}, wechselt er in den Zustand Ceremony.	Client, ClientState, ClientGameLogic, Game, Turn, MovePiece, Ceremony	JUnit	erfolgreich	
T201	UC-ClientState-41	testStayInChoosePowerCard	Der Client bleibt im Zustand ChoosePowerCard, erhält Nachrichten und ändert den State nicht	Client, ClientState, ClientGameLogic, Game, Turn, PowerCard, ChoosePowerCard	JUnit	erfolgreich	
T202	UC-ClientState-42	testChoosePowerCardToRollDice	Soll ohne eine Bonuskarte zu spielen gewürfelt werden, wechselt der Client in den Zustand RollDice.	Client, ClientState, ClientGameLogic, Game, Turn, PowerCard, ChoosePowerCard	JUnit	erfolgreich	
T203	UC-ClientState-43	testChoosePowerCardToSwap	Der Spieler bekommt die Message {SelectPieces} und wechselt in Swap	Client, ClientState, ClientGameLogic, Game, Turn, PowerCard, ChoosePowerCard, Swap	JUnit	erfolgreich	
T204	UC-ClientState-44	testChoosePowerCardToShield	Der Benutzer erhält die Message {SelectPieces} und wechselt in Shield	Client, ClientState, ClientGameLogic, Game, Turn, PowerCard, ChoosePowerCard, Shield	JUnit	erfolgreich	
T205	UC-ClientState-45	testStayInShield	Der Benutzer bekommt Messages und verlässt diesen State nicht.	Client, ClientState, ClientGameLogic, Game, Turn, PowerCard, Shield	JUnit	erfolgreich	
T206	UC-ClientState-46	testShieldToPowerCardEndState	Wurde das Spielen der Schild-Karte durch den Server validiert, wechselt der Client in den EndState von PowerCard.	Client, ClientState, ClientGameLogic, Game, Turn, PowerCard, Shield	JUnit	erfolgreich	
T207	UC-ClientState-47	testSwapToPowerCardEndState	Wurde das Spielen der Swap-Karte durch den Server validiert, wechselt der Client in den EndState von PowerCard.	Client, ClientState, ClientGameLogic, Game, Turn, PowerCard, ChoosePowerCard, Swap	JUnit	erfolgreich	
T208	UC-ClientState-48	testNoPieceInWaitingPiece	Der Client geht von NoPiece in WaitInPiece (in ChoosePiece) mit der Message {WaitingPiece}.	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, NoPiece, WaitingPiece	JUnit	erfolgreich	
T209	UC-ClientState-49	testNoPieceInSelectedPiece	Der Client geht von NoPiece in SelectPiece (in ChoosePiece) mit der Message {SelectPiece}	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, NoPiece, SelectPiece	JUnit	erfolgreich	
T210	UC-ClientState-50	testNoPieceInStartPiece	Der Client geht von NoPiece in StartPiece (in ChoosePiece) mit der Message {StartPiece}	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, NoPiece, StartPiece	JUnit	erfolgreich	
T211	UC-ClientState-51	testNoPieceInWait	Der Client geht von NoPiece in Wait (in ChoosePiece) mit der Message {NoTurn}	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, NoPiece, Wait	JUnit	erfolgreich	
T212	UC-ClientState-52	testStayInWaitingPiece	Der Client ist in WaitingPiece und erhält Messages und wechselt den State nicht	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, WaitingPiece	JUnit	erfolgreich	
T213	UC-ClientState-53	testWaitingPieceInChoosePieceEndState	Der Client ist in WaitingPiece und erhält die Nachricht {MoveMessage} und geht in den Endzustand von ChoosePiece	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, WaitingPiece	JUnit	erfolgreich	
T214	UC-ClientState-54	testStayInSelectedPiece	Der Client ist in SelectPiece und erhält Messages und wechselt den State nicht	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, SelectPiece	JUnit	erfolgreich	
T215	UC-ClientState-55	testSelectedPieceInChoosePieceEndState	Der Client ist in SelectPiece und erhält die Nachricht {MoveMessage} und geht in den Endzustand von ChoosePiece	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, SelectPiece	JUnit	erfolgreich	
T216	UC-ClientState-56	testStayInStartPiece	Der Client ist in StartPiece und erhält Messages, wechselt den State aber nicht	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, StartPiece	JUnit	erfolgreich	
T217	UC-ClientState-57	testStartPieceToChoosePieceEndState	Der Client ist in StartPiece und erhält die Nachricht {MoveMessage} und geht in den Endzustand von ChoosePiece	Client, ClientState, ClientGameLogic, Game, Turn, ChoosePiece, StartPiece	JUnit	erfolgreich	
T218	UC-ClientState-58	testRollRankingDiceToWaitRanking	Nach dem Würfeln um die Zugreihenfolge, wartet der Client im Zustand WaitRanking auf die ServerMessage {RankingResponse} oder {RankingRollAgain}.	Client, ClientState, ClientGameLogic, Game, Turn, DetermineStartPlayer, RollRankingDice, WaitRanking	JUnit	erfolgreich	
T219	UC-ClientState-59	testWaitRankingToRollRankingDice	Muss erneut gewürfelt werden, wechseln die Clients, die die selbe Augenzahl gewürfelt haben wieder in den Zustand RollRankingDice.	Client, ClientState, ClientGameLogic, Game, Turn, DetermineStartPlayer, RollRankingDice, WaitRanking	JUnit	erfolgreich	
T220	UC-ClientState-60	testWaitRankingToEndStateDetermineStartingPlayer	Steht eine Zugreihenfolge fest, wechseln die Clients in den EndState von DetermineStartingPlayer.	Client, ClientState, ClientGameLogic, Game, Turn, DetermineStartPlayer, WaitRanking	JUnit	erfolgreich	
T221	UC-ClientState-61	testPodiumToStatistics	Nach der Siegerehrung wechselt der Client zur Darstellung der Statistiken in den Zustand Statistics.	Client, ClientState, ClientGameLogic, Ceremony, Statistics, Podium	JUnit	erfolgreich	
T222	UC-ClientState-62	testStatisticsToCeremonyEndState	Wird der Statistikfenster verlassen, wechselt der Client in den EndState von Ceremony.	Client, ClientState, ClientGameLogic, Ceremony, Statistics	JUnit	erfolgreich	