Dokumentation

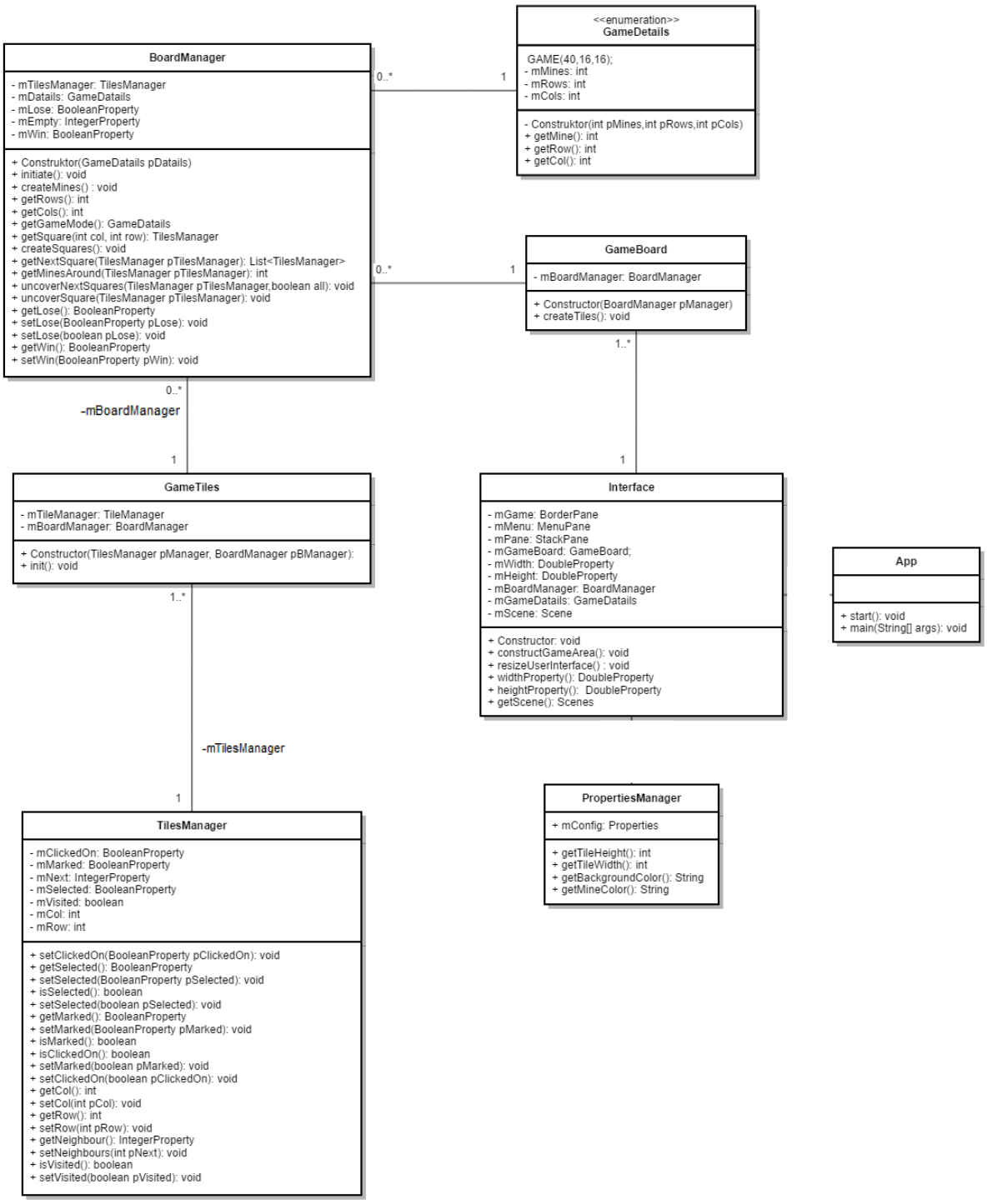
MODUL 266A

Nana Schütz
INF15A | CSBE

Inhaltsverzeichnis

Dokumentation – Mine Sweeper

Klassendiagramm



Fazit

Es war ein spannendes Projekt und ebenfalls hatte ich Spass beim Programmieren gehabt. Doch dies war jedoch nicht immer der Fall, da dieses Projekt mir viel Zeit und Nerven gekostet hat. Ich wurde nämlich nicht wie geplant fertig und hatte sehr viel Zeit mit überlegen, wie ich die Anforderungen lösen soll verbracht. Was mir ebenfalls viel Zeit gekostet hat, war das Recherchieren, denn ich habe viele Tutorials angeschaut, damit ich auch alles was ich ins Programm reintue auch verstehe. Am Schluss konnte ich jedoch eine einiger Masen gute und lauffähige Applikation abgeben.

Dokumentation – Hangman

Funktionale Anforderungen

UC-ID	1
Name	Key-Erkennung
Akteure	Player
Normaler Ablauf	Alle gedrückte Buchstaben-Testen werden erkannt
Alternativer Ablauf	Alle gedrückte Buchstaben-Testen werden nicht erkannt/ Einige Tasten werden nicht erkannt
Vorbedingung	-
Nachbedingung	UC-ID 2

UC-ID	2
Name	Streichung der Buchstaben
Akteure	Player
Normaler Ablauf	Bereits benutzte Buchstaben werden gestrichen
Alternativer Ablauf	Benutzte Buchstaben werden nicht erkannt/ Einige benutzte Buchstaben werden nicht gestrichen
Vorbedingung	UC-ID 1
Nachbedingung	-

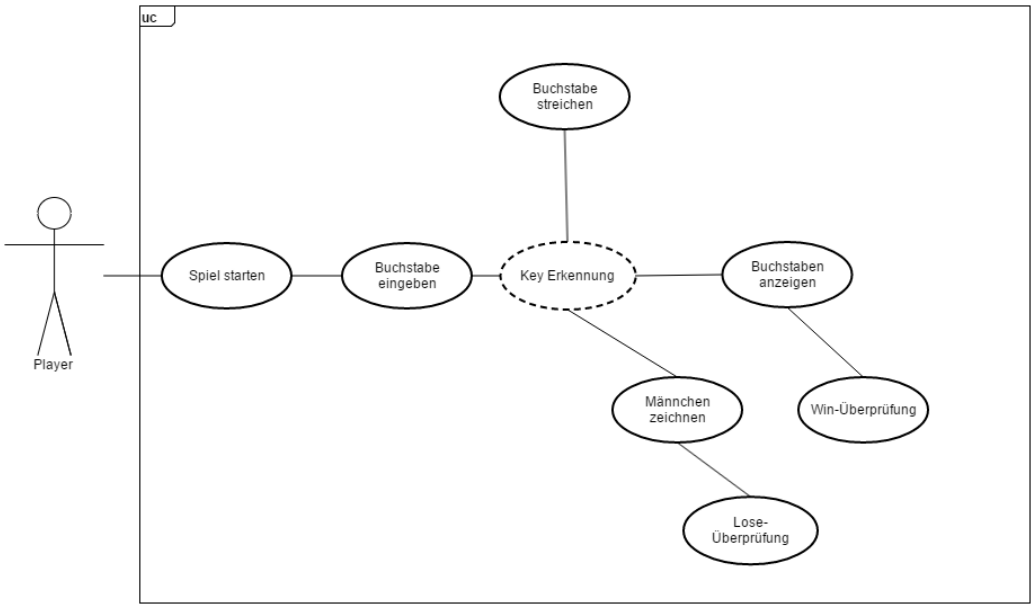
UC-ID	3
Name	Buchstaben anzeigen
Akteure	Player
Normaler Ablauf	Korrekte Buchstaben werden an richtiger Position angezeigt
Alternativer Ablauf	Korrekte Buchstaben werden nicht angezeigt/ Einige korrekte Buchstaben werden nicht angezeigt
Vorbedingung	UC-ID 1
Nachbedingung	UC-ID 5

UC-ID	4
Name	Männchen zeichnen
Akteure	Player
Normaler Ablauf	Pro falsche Eingabe wird ein Strich gezeichnet
Alternativer Ablauf	Pro falsche Eingabe wird kein Strich gezeichnet/ Pro falsche Eingabe werden mehrere Striche gezeichnet
Vorbedingung	UC-ID 1
Nachbedingung	UC-ID 6

UC-ID	5
Name	
Akteure	Player
Normaler Ablauf	Der Player gewinnt, wenn alle Buchstaben aufgedeckt sind
Alternativer Ablauf	Der Player verliert, wenn alle Buchstaben aufgedeckt sind/ Keine Aktion nachdem alle Buchstaben aufgedeckt sind.
Vorbedingung	UC-ID 3
Nachbedingung	-

UC-ID	6
Name	Lose-Überprüfung
Akteure	Player
Normaler Ablauf	Der Player verliert, wenn das Männchen fertiggezeichnet wurde
Alternativer Ablauf	Der Player gewinnt, wenn das Männchen fertig gezeichnet wurde/ keine Aktion nachdem das Männchen fertig gezeichnet wurde
Vorbedingung	UC-ID 4
Nachbedingung	-

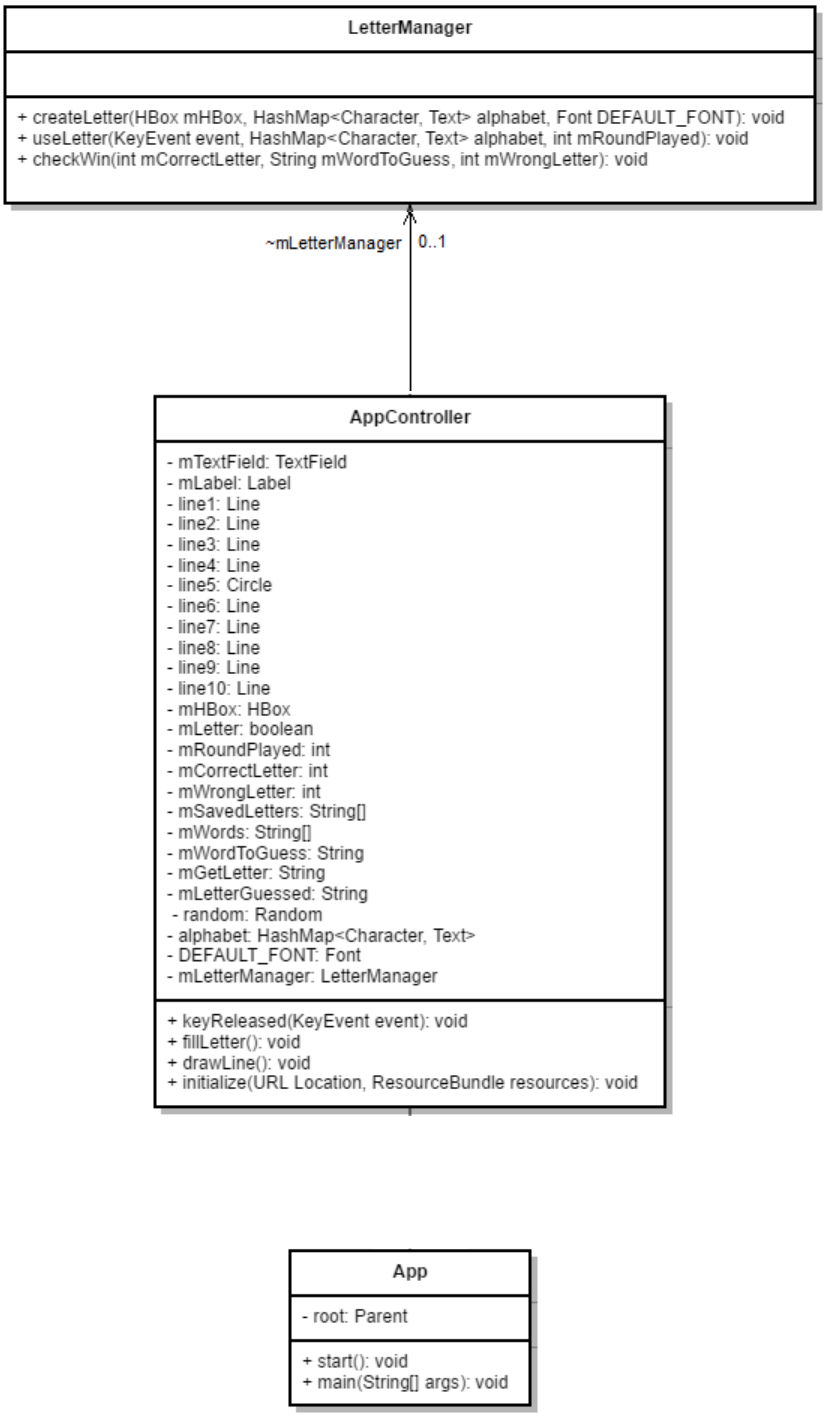
Use-Case-Diagramm



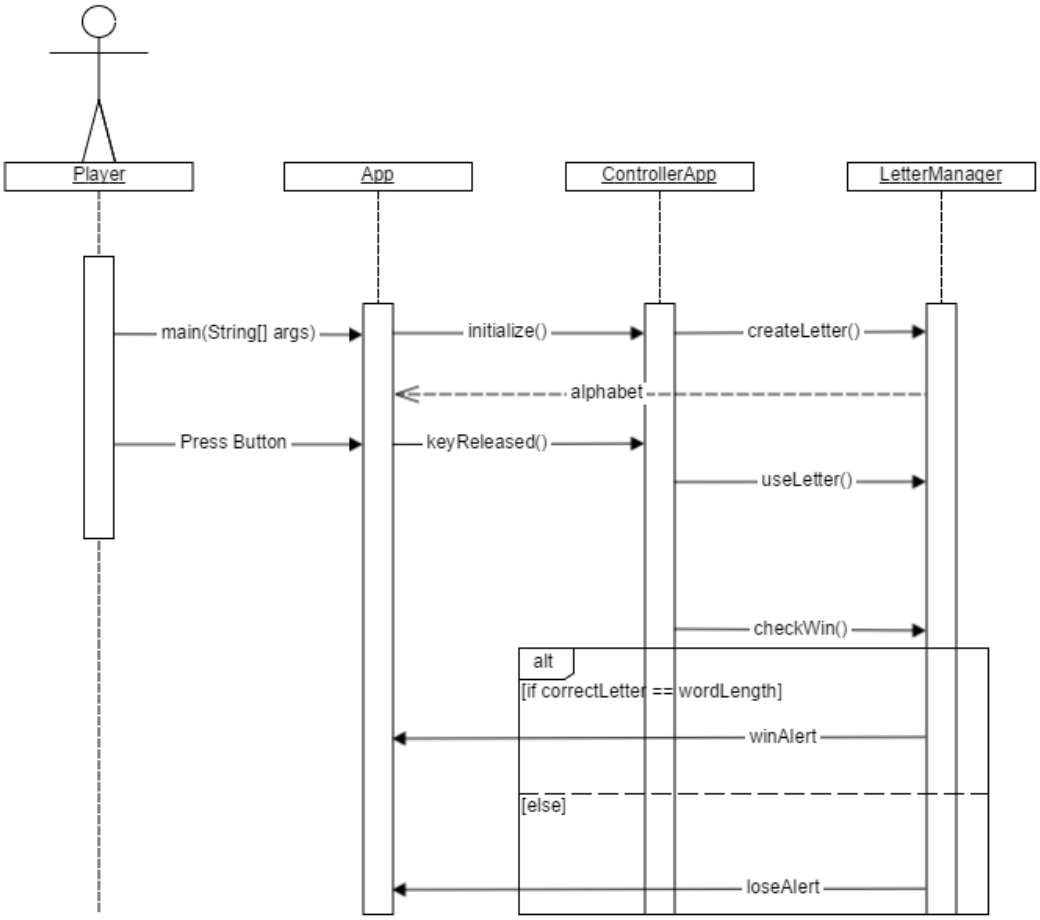
Nicht funktionale Anforderungen

NFA		Wichtigkeit		
Anforderung	**Messwert**	**Hoch**	**Mittel**	**Niedrig**
Technologie	GUI mit JavaFX implementiert	x		
Benutzbarkeit	Startwartezeit geringer als 1 Sekunde		x	
Speicherplatz	< 30 MB	x		
Umgebung	Mit Eclipse entwickelt	x		
Fehlertoleranz	Fehlern bringen die Applikation nicht zum Absturz	x		

Klassendiagramm



Sequenzdiagramm



Testing

WhiteBox

Tester: Nana Schütz

Test-ID	001
UC-ID	1
Beschreibung	Key-Erkennung
Testvoraussetzung	-
Testmethode	White-Box
Priorität	hoch
Testablauf	Der Tester drückt auf einige Buchstaben-Tasten
Erwartetes Ergebnis	Alle gedrückte Buchstaben-Testen werden erkannt
Test-Ergebnis	Bestanden

Test-ID	002
UC-ID	2
Beschreibung	Streichung der Buchstaben
Testvoraussetzung	UC-ID 1
Testmethode	White-Box
Priorität	hoch
Testablauf	Der Tester drückt auf einige beliebige Buchstaben-Tasten
Erwartetes Ergebnis	Bereits benutzte Buchstaben werden gestrichen
Test-Ergebnis	Bestanden

Test-ID	003
UC-ID	3
Beschreibung	Buchstaben anzeigen
Testvoraussetzung	UC-ID 1
Testmethode	White-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf einige korrekte Buchstaben
Erwartetes Ergebnis	Korrekte Buchstaben werden an richtiger Position angezeigt
Test-Ergebnis	Bestanden

Test-ID	004
UC-ID	4
Beschreibung	Männchen zeichnen
Testvoraussetzung	UC-ID 1
Testmethode	White-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf einige falsche Buchstaben
Erwartetes Ergebnis	Pro falsche Eingabe wird ein Strich gezeichnet
Test-Ergebnis	Bestanden

Test-ID	005
UC-ID	5
Beschreibung	Win-Überprüfung
Testvoraussetzung	UC-ID 3
Testmethode	White-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf alle richtigen Buchstaben
Erwartetes Ergebnis	Der Player gewinnt, wenn alle Buchstaben aufgedeckt sind
Test-Ergebnis	Bestanden

Test-ID	006
UC-ID	6
Beschreibung	Lose-Überprüfung
Testvoraussetzung	UC-ID 4
Testmethode	White-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf einige falsche Buchstaben bis das Männchen fertig gezeichnet wurde
Erwartetes Ergebnis	Der Player verliert, wenn das Männchen fertiggezeichnet wurde
Test-Ergebnis	Bestanden

BlackBox

Tester: Alberto Schütz

Test-ID	001
UC-ID	1
Beschreibung	Key-Erkennung
Testvoraussetzung	-
Testmethode	Black-Box
Priorität	hoch
Testablauf	Der Tester drückt auf einige Buchstaben-Tasten
Erwartetes Ergebnis	Alle gedrückte Buchstaben-Testen werden erkannt
Test-Ergebnis	Bestanden

Test-ID	002
UC-ID	2
Beschreibung	Streichung der Buchstaben
Testvoraussetzung	UC-ID 1
Testmethode	Black-Box
Priorität	hoch
Testablauf	Der Tester drückt auf einige beliebige Buchstaben-Tasten
Erwartetes Ergebnis	Bereits benutzte Buchstaben werden gestrichen
Test-Ergebnis	Bestanden

Test-ID	003
UC-ID	3
Beschreibung	Buchstaben anzeigen
Testvoraussetzung	UC-ID 1
Testmethode	Black-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf einige korrekte Buchstaben
Erwartetes Ergebnis	Korrekte Buchstaben werden an richtiger Position angezeigt
Test-Ergebnis	Bestanden

Test-ID	004
UC-ID	4
Beschreibung	Männchen zeichnen
Testvoraussetzung	UC-ID 1
Testmethode	Black-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf einige falsche Buchstaben
Erwartetes Ergebnis	Pro falsche Eingabe wird ein Strich gezeichnet
Test-Ergebnis	Bestanden

Test-ID	005
UC-ID	5
Beschreibung	Win-Überprüfung
Testvoraussetzung	UC-ID 3
Testmethode	Black-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf alle richtigen Buchstaben
Erwartetes Ergebnis	Der Player gewinnt, wenn alle Buchstaben aufgedeckt sind
Test-Ergebnis	Bestanden

Test-ID	006
UC-ID	6
Beschreibung	Lose-Überprüfung
Testvoraussetzung	UC-ID 4
Testmethode	Black-Box
Priorität	hoch
Testablauf	Der Tester bekommt das Wort vorgegeben und drückt dann auf einige falsche Buchstaben bis das Männchen fertig gezeichnet wurde
Erwartetes Ergebnis	Der Player verliert, wenn das Männchen fertiggezeichnet wurde
Test-Ergebnis	Bestanden

Fazit

Das Hangman-Programm fiel mir wesentlich leichter als den Mine Sweeper. Jedoch war ich krank und konnte die Applikation nicht so erweitern und gestalten wie ich es machen wollte. Doch schlussendlich bin ich mit diesem Projekt zufrieden, da ich es innerhalb von 8 h geschrieben habe.

Ich habe in diesem Modul viel Neues über Java gelernt, vor allem wo ich den MineSweeper geschrieben habe.