

Testplan zu Patchbot – The Game

Testgebiete (Struktur):

Tests 1 bis 16 : Funktionalität der GUI Elemente
 Tests 17 bis 19 : Einlesen von Dateien
 Tests 20 bis 32 : Funktionalität der Rendering Engine
 Tests 33 bis 43 : Interaktionen zwischen Umgebung und Patchbot
 Tests 44 bis 50 : Gegnerische Roboter.
 Tests 51 bis 53 : Last Tests

Nr.	Voraussetzung	Durchführung	Erwartung	Ergebnis
0	Patchbot installiert	Patchbot starten	Fenster mit erster Kolonie (hunted) erscheint.	OK
1	Test #0 durchgeführt	Fenster maximieren klicken	Fenster füllt den gesamten Bildschirm aus und die GUI und Karte werden korrekt angezeigt.	OK
2	Test #0 durchgeführt	Fenster schließen klicken oder Alt + F4 drücken	Das Programm beendet.	OK
3	Test #0 durchgeführt	Schaltfläche „Andere Kolonie...“ klicken	Ein Dateiauswahlfenster öffnet sich.	OK
4	Test #3 durchgeführt	„test_maps/test_1.txt“ auswählen und bestätigen	Das Dateiauswahlfenster schließt sich und die Karte „test_1“ wird angezeigt. Außerdem wird dieser Name oben links angezeigt.	OK
5	Test #4 durchgeführt	Wiederholungen auf „Bis Hindernis“ einstellen und Pfeil nach rechts anklicken.	Im Programmierfeld steht „RX“.	OK
6	Test #5 durchgeführt	Wiederholungen auf 2 einstellen und Pfeil nach unten klicken.	Im Programmierfeld steht „RXD2“.	OK
7	Test #6 durchgeführt	Test #6 wiederholen.	Im Programmierfeld steht „RXD4“.	OK
8	Test #7 durchgeführt	Wiederholungen auf 10 einstellen und den mittigen Knopf (Warte-Knopf) klicken	Im Programmierfeld steht „RXD4W10“.	OK
9	Test #8 durchgeführt	Zurück-Knopf klicken.	Im Programmierfeld steht „RXD4“.	OK
10	Test #0 durchgeführt	Wiederholung „Bis Hindernis“ auswählen und den Warte-Knopf klicken.	Fehler, da der Warte-Befehl inkompatibel ist mit „Bis Hindernis“.	OK
11	Test #0 durchgeführt	Ca. 10-mal abwechselnd 2 oder mehr beliebige Pfeile klicken.	Die komplette Befehlsfolge passt nicht in das Programmierfeld, und die Scroll Bar unter dem Programmierfeld lässt sich bewegen und sich damit die gesamte Befehlsfolge lesen.	OK

12	Test #9 durchgeführt	Start-Knopf klicken.	<ul style="list-style-type: none"> - Abbruch-, Einzelschritt- und Automatik-Knopf werden aktiviert und der Start-Knopf deaktiviert. - Alle Bedienelemente im Bereich „Programmieren“ und die Schaltfläche „Andere Kolonie...“ werden deaktiviert. - Alle Roboter erscheinen auf der Karte und Pfeile erscheinen, die den schnellsten Weg zu Patchbot zeichnen. 	OK
13	Test #12 durchgeführt	Einzelschritt-Knopf klicken.	Patchbot bewegt sich um 1 Feld nach rechts.	OK
14	Test #12 durchgeführt	Automatik-Knopf klicken.	Der Anhalten-Knopf wird aktiviert, Einzelschritt- und Automatik-Knopf deaktiviert und Patchbot fängt an sich nach rechts zu bewegen.	OK
15	Test #14 durchgeführt	Anhalten-Knopf klicken.	Der Anhalten-Knopf wird deaktiviert, Einzelschritt- und Automatik-Knopf aktiviert und Patchbot hört auf sich zu bewegen.	OK
16	Test #12 durchgeführt	Abbruch-Knopf klicken oder Befehlsfolge zu Ende laufen lassen.	<ul style="list-style-type: none"> - Das Spiel ist verloren. Dies wird durch ein Dialogfenster propagiert. - Alle Knöpfe im Bereich „Missionsablauf“ werden deaktiviert, der Start-Knopf wird aktiviert. - Alle Bedienelemente im Bereich „Programmieren“ und die Schaltfläche „Andere Kolonie...“ werden aktiviert. - Alle Roboter und die Pfeile verschwinden von der Karte. - Befehlsfolge ist erhalten und kann erneut genutzt werden. 	OK
17	Test #16 durchgeführt	Schaltfläche „Andere Kolonie...“ klicken, nichts auswählen und Dateiauswahlfenster schließen.	Es wird durch ein Dialogfenster propagiert, dass der Pfad nicht valide ist.	OK
18	Test #16 durchgeführt	Jede Karte im Ordner „broken_maps“ öffnen.	Keine Ladeversuch verändert die Umgebung, da jede Karte beschädigt ist. Stattdessen wird bei jeder Karte durch ein Dialogfenster propagiert, warum der Ladeversuch fehlschlug, genauer, was an der Karte beschädigt ist.	OK
19	Test #17 und Test#18 durchgeführt	Wiederhole Tests Nr. 5, 6, 7, 12, 13, 14, 15, 16.	Die Fehlgeschlagenen Ladeversuche in Test #17 und Test #18 haben keine Auswirkungen auf das Programm. Alle Tests führen das gleiche Ergebnis herbei,	OK

			wie vor der Durchführung von Test #17 und Test #18.	
20	Test #0 durchgeführt	Karte „test_maps/check_all_tiles.txt“ einlesen und Start-Knopf drücken.	Diese Karte enthält eins von jedem möglichen Feld und Roboter-Startpunkt und zweimal Stahlplankenboden, d.h. es gibt (neben den Stahlplanken) keine doppelten Felder oder Roboter, woraus sich schließen lässt, dass das Einlesen der Felder funktioniert.	OK
21	Test #0 durchgeführt	Karte „everything.txt“ einlesen.	Ein Ausschnitt der Karte „everything“ wird angezeigt. Die Scrollbalken sind aktiviert.	OK
22	Test #21 durchgeführt	Fenster vergrößern. (Größer und kleiner „ziehen“)	Der Kartenausschnitt wird weich auf Pixelbasis angepasst und neu gerendert.	OK
23	Test #21 durchgeführt	Fenster über die Breite und Höhe der Karte hinaus vergrößern.	Der Kartenausschnitt wird korrekt dargestellt, darüber hinaus wird Whitespace dargestellt. (nichts)	OK
24	Test #21 durchgeführt	Horizontal und vertikal Scrollen.	Die Scrollbalken verschieben den Kartenausschnitt korrekt.	OK
25	Test #21 durchgeführt	Beide Scrollbalken mit zum Maximum verschieben.	Es wird ein Kartenausschnitt angezeigt, indem das Feld in der rechten unteren Ecke dem Feld in der rechten unteren Ecker der Karte entspricht.	OK
26	Test #25 durchgeführt	Während bis zum Maximum gescrollt ist, Fenster vergrößern.	Es wird kein Whitespace rechts von der Karte angezeigt, stattdessen „klebt“ die Karte am rechten Rand und der Kartenausschnitt wird so lange vergrößert bis auch der linke Rand zu sehen ist; erst dann wird Whitespace rechts von der Karte sichtbar. Scrollbalken sind korrekt angepasst.	OK
27	Test #21 durchgeführt	Fenster maximieren.	Es wird der korrekte Kartenausschnitt angezeigt, ggf. mit Whitespace, wenn das Fenster größer ist als die Karte. Scrollbalken sind korrekt angepasst.	OK
28	Test #25 durchgeführt	Während bis zum Maximum gescrollt ist, Fenster maximieren.	Es wird der korrekte Kartenausschnitt angezeigt, ggf. mit Whitespace, wenn das Fenster größer ist als die Karte. Scrollbalken sind korrekt angepasst.	OK
29	Test #21 durchgeführt	Das Fenster an der unteren rechten Ecke zum Vergrößern „greifen“ und wild und rasch die Maus bewegen.	Es treten keine Glitches auf, d.h. es wird immer noch der korrekte Kartenausschnitt dargestellt und das Programm stürzt nicht ab. Zudem sind die Scrollbalken korrekt angepasst.	OK
30	Test #21 durchgeführt	Beide Scrollbalken „greifen“ und die Maus rasch auf und ab, bzw. hin und her bewegen.	Es treten keine Glitches auf, d.h. es wird immer noch der korrekte	OK

			Kartenausschnitt dargestellt und das Programm stürzt nicht ab.	
31	Test #0 durchgeführt	Karte „test_maps/large“ (10.000x10.000 Tiles) öffnen.	Ein Ausschnitt der Karte „large“ wird angezeigt. Die Scrollbalken sind aktiviert und korrekt angepasst.	OK
32	Test #31 durchgeführt	Fenster vergrößern. (Größer und kleiner „ziehen“)	Der Kartenausschnitt wird weich auf Pixelbasis angepasst und neu gerendert. Es sind trotz der Größe der Karte keine signifikanten Performanceeinbußen zu verzeichnen.	OK
33	Test #0 durchgeführt	Karte „test_maps/interaction_tests_patchbot.txt“ laden.	Karte wird korrekt geladen.	OK
34	Test #33 durchgeführt	Befehlsfolge „UX“ ausführen. ¹	Patchbot läuft auf Abgrund zu und stirbt. Das Spiel ist verloren, was durch ein Dialogfenster propagiert wird.	OK
35	Test #33 durchgeführt	Befehlsfolge „R1UX“ asuführen. ¹	Patchbot läuft auf Wasser zu und stirbt. Das Spiel ist verloren, was durch ein Dialogfenster propagiert wird.	OK
36	Test #33 durchgeführt	Befehlsfolge „R2U1“ asuführen. ¹	Patchbot läuft auf den Hauptserver zu und gewinnt das Spiel, was durch ein Dialogfenster propagiert wird.	OK
37	Test #33 durchgeführt	Befehlsfolge „D4“ ausführen. ¹	Patchbot läuft auf den Rand der Karte zu, bleibt aber auf der Karte. Kein Fehlverhalten.	OK
38	Test #33 durchgeführt	Befehlsfolge „R1D1R1D1“ ausführen. ¹	Patchbot läuft zuerst auf eine Betonwand zu und dann auf eine Felswand. Beide blockieren ihn. Weiter passiert nichts.	OK
39	Test #33 durchgeführt	Befehlsfolge „R3D2U1W8“ ausführen. ¹	Patchbot läuft auf eine Manuelle Tür zu und öffnet sie damit (Zeitschritt 1), dann läuft er in/auf die offene Tür (Zeitschritt 2), dann geht er wieder ein Feld zurück (Zeitschritt 3) und wartet 8 Zeitschritte. Da eine Tür für 10 Zeitschritte offen ist, schließt sich die Tür mit dem 8ten Warten-Befehl wieder.	
40	Test #33 durchgeführt	Befehlsfolge „R3D2W13U1“ ausführen. ¹	Patchbot öffnet wieder die Manuelle Tür, wie in Test #39, nur bleibt er diesmal für 13 Zeitschritte in der Tür stehen. Sie bleibt länger als die normalen 10 Zeitschritte blockiert und schließt, wenn Patchbot sie zum Schluss verlässt.	OK

¹ Richtung + „X“ bedeutet „Bis Hindernis“, Richtung + Zahl bedeutet „wiederhole n-mal“. Zum Ausführen Start- und dann Automatikknopf drücken. (U: Pfeil nach oben; R: Pfeil nach rechts; D: Pfeil nach unten; L: Pfeil nach links)

41	Test #33 durchgeführt	Befehlsfolge „R4D1“ ausführen. ¹	Patchbot läuft auf eine, für ihn nicht zu öffnende, Automatische Türe zu. Es passiert nichts.	OK
42	Test #33 durchgeführt	Befehlsfolge „R3U1“ ausführen. ¹	Patchbot läuft in einen, nur für ihn zugänglichen Geheimgang.	OK
43	Test #33 durchgeführt	Befehlsfolge „RX“ ausführen. ¹	Patchbot läuft bis zum rechten Kartenrand. Dabei überquert er Schotter, das ihn nicht verlangsamt, und Aliengras, das ihn für einen Zeitschritt blockiert.	OK
44	Test #0 durchgeführt	Karte „test_maps/bugger_test_map.txt“ öffnen und Befehlsfolge „W39“ ausführen.	Bugger folgt der Wand im Uhrzeigersinn, bis er wieder an seinem Startpunkt angelangt ist und sich eine neue Wand sucht. Von dort aus startet er eine neue „Runde“ bis er wieder am Start ist und wieder eine neue Wand sucht.	OK
45	Test #0 durchgeführt	Karte „test_maps/pusher_test_map.txt“ öffnen und Befehlsfolge „W15“ ausführen.	Pusher bewegt sich horizontal bis er Patchbots Breite erreicht hat und dann vertikal in Richtung Patchbot, welchen er dann auch bis zum unteren Rand verschiebt, ohne ihn in die Wand zu schieben.	OK
46	Test #0 durchgeführt	Karte „test_maps/digger_test_map.txt“ öffnen und Befehlsfolge „W15“ ausführen.	Digger verhält sich wie Pusher in Test #45, mit dem Unterschied, dass er auf seinem Weg eine Felswand hat, die er zu Schotter verwandelt.	OK
46	Test #0 durchgeführt	Karte „test_maps/swimmer_test_map.txt“ öffnen und Befehlsfolge „W15“ ausführen.	Swimmer verhält sich wie Pusher in Test #45, mit dem Unterschied, dass er auf seinem Weg Feld Wasser hat, das er überquert.	OK
47	Test #0 durchgeführt	Karte „test_maps/follower_test_map.txt“ öffnen und Befehlsfolge „DXR1UX“ ausführen.	Sobald der Follower Patchbot sieht verfolgt er ihn. Wenn er ihn erreicht, ist das Spiel verloren.	OK
48	Test #0 durchgeführt	Karte „test_maps/hunter_test_map.txt“ öffnen und Befehlsfolge „DXR1UX“ ausführen.	Sobald der Hunter Patchbot gesehen hat verfolgt er ihn mit doppelten Schritten, auch wenn Patchbot wieder außer Sicht ist. Wenn er ihn erreicht, ist das Spiel verloren.	OK
49	Test #0 durchgeführt	Karte „test_maps/sniffer_test_map.txt“ öffnen und Befehlsfolge „DXRX“ ausführen.	Der Sniffer fängt sofort an in Richtung Patchbot zu laufen, da er ihn auch verfolgt, ohne ihn zu sehen. Wenn er ihn erreicht, ist das Spiel verloren.	OK
50	Test #0 durchgeführt	Karte „everything.txt“ öffnen und die Befehlsfolge „LX U1 L3 UX L5 D3 L8 U2 R5 U4 L5 UX L1 UX R23 U5 R3 UX D1 RX“ ausführen.	Patchbot gewinnt das Spiel.	OK
51	Test #0 durchgeführt	Karte „test_maps/large.txt“ (10.000 * 10.000 Tiles) laden.	Laden der Karte dauert länger als „normal“. Performanz während dem Spiel	OK

			nicht schlechter als bei kleineren Karten. (s. Test #31)	
			<ul style="list-style-type: none"> ➔ Um hier beim Laden besser Zeiten zu erreichen, bietet es sich an ggf. die Einlese Methode effizienter zu gestalten, bzw. dort Optimierungen vorzunehmen. 	
52	Test #0 durchgeführt	Karte „test_maps/many_robots.txt“ (2240 Roboter) öffnen und die Befehlsfolge „W5“ ausführen.	<p>Schlechte Performance da die State Machines sequenziell ausgeführt werden.</p> <ul style="list-style-type: none"> ➔ Ein Lösungsansatz wäre hier die parallele Verarbeitung der State Machines, wobei auch da der Speed-up nicht allzu immens wäre, da man die map mit Mutex sperren muss um einen konsistenten Kartenzustand zu Behalten. 	OK
53	Noch kein Test durchgeführt	Ersetze Umgebungsbilder im Ordner Umgebungen durch größere Bilder (2.000x2.000px) und starte das Programm	<p>Längere Ladezeiten und höherer RAM Verbrauch.</p> <ul style="list-style-type: none"> ➔ Als einen primitiven Lösungsversuch kann man hier versuchen die Bild Auflösung intern zu verringern, da die Größen der Tiles ohnehin nicht von 2.000x2.000px oder mehr profitieren können. 	