# Bugger

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | FW | = Folge Wand | | W | = Warte | | FaW | = Finde andere Wand | | |  |  | | --- | --- | | zb\_u\_ns | = Zielfeld belegt u. nicht Start? | | zf | = Zielfeld frei? | | zb | = Zielfeld belegt? | | zf\_u\_ns | = Zielfeld frei u. nicht Start? | | zs | = Zielfeld Start? | | zw | = Zielfeld Wand? | | znw\*1) | = Zielfeld nicht Wand? | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | zb\_u\_ns | zf | zb | zf\_u\_ns | zs | zw | znw |
| * FW\* | W | - | - | FW | FaW | - | - |
| W | - | FW | W | - | - | - | - |
| FaW\*) | - | - | - | - | - | FW | FaW |

\*) Bei Zustand nochmal Event prüfen um richtiges Zielfeld gespeichert zu haben  
\*1) Da die Graphen in der Aufgabenstellung so schlecht durchdacht sind, findet der Check, ob das Zielfeld frei ist in der state\_func zu FaW statt

# Pusher[[1]](#footnote-1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | HB | = Horizontale Bewegung | | VB | = Vertikale Bewegung | | Z | = Zerstört | | |  |  | | --- | --- | | zf | = Zielfeld frei (Oder Roboter, aber verschiebbar)? | | zw\_o\_pbb | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Breite erreicht? | | zw\_o\_pbh | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Höhe erreicht? | | zg | = Zielfeld gefährlich? | |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | zf | zw\_o\_pbb | zw\_o\_pbh | zg |
| * HB | HB | VB | - | Z |
| VB | VB | - | HB | Z |
| Z | - | - | - | - |

# Digger1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | HB | = Horizontale Bewegung | | VB | = Vertikale Bewegung | | Z | = Zerstört | | |  |  | | --- | --- | | zf | = Zielfeld frei (Oder Roboter, aber verschiebbar)? | | zw\_o\_pbb | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Breite erreicht? | | zw\_o\_pbh | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Höhe erreicht? | | zg | = Zielfeld gefährlich? | |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | zf | zw\_o\_pbb | zw\_o\_pbh | zg |
| * HB | HB | VB | - | Z |
| VB | VB | - | HB | Z |
| Z | - | - | - | - |

# Swimmer[[2]](#footnote-2)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | HB | = Horizontale Bewegung | | VB | = Vertikale Bewegung | | Z | = Zerstört | | |  |  | | --- | --- | | zf\_o\_w | = Zielfeld frei (Oder Roboter, aber verschiebbar) o. Wasser? | | zw\_o\_pbb | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Breite erreicht? | | zw\_o\_pbh | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Höhe erreicht? | | zg | = Zielfeld gefährlich? | |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | zf\_o\_w | zw\_o\_pbb | zw\_o\_pbh | zg |
| * HB | HB | VB | - | Z |
| VB | VB | - | HB | Z |
| Z | - | - | - | - |

# Pushing Robot2

(Pushing Robot sind alle „pushing robots” zusammengefasst. So wurden sie auch implementiert. Roboter-spezifische Aktionen (z.B. schwimmen und graben, wurden dort konditional mit eingebaut.)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | HB | = Horizontale Bewegung | | VB | = Vertikale Bewegung | | Z | = Zerstört | | |  |  | | --- | --- | | zf | = Zielfeld frei (Oder Roboter, aber verschiebbar) o. Wasser? | | zw\_o\_pbb | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Breite erreicht? | | zw\_o\_pbh | = Zielfeld Wand (o. block. Rob.) o. Patchbot-Höhe erreicht? | | zg | = Zielfeld gefährlich? | |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | zf\_o\_w | zw\_o\_pbb | zw\_o\_pbh | zg |
| * HB | HB | VB | - | Z |
| VB | VB | - | HB | Z |
| Z | - | - | - | - |

# Follower[[3]](#footnote-3)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | V | = Verfolge | | W | = Warte | |  |  | | |  |  | | --- | --- | | pb\_is | = Patchbot in Sicht und erreichbar? | | pb\_as | = Patchbot außer Sicht oder nicht erreichbar? | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | pb\_is | pb\_as |
| * W | V | W |
| V | V | W |

# Hunter3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | V | = Verfolge | | W | = Warte | | J | = Jage | |  |  | | |  |  | | --- | --- | | pb\_is | = Patchbot in Sicht und erreichbar? | | pb\_as | = Patchbot außer Sicht oder nicht erreichbar? | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | pb\_is | pb\_as |
| * W | V | - |
| V | V | J |
| J | V | J |

# Sniffer

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | V | = Verfolge | | W | = Warte | |  |  | | |  |  | | --- | --- | | pe | = Patchbot erreichbar? | | pne | = Patchbot nicht erreichbar? | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | pe | pne |
| * W | V | W |
| V | V | W |

*Follower, Hunter und Sniffer zusammenzufassen gestaltet sich etwas schwieriger, zumindest als Tabelle hier. Bei der Implementierung bietet es sich an, um trotzdem alle drei FSMs in einer Klasse umzusetzen, eine map<robot\_type, fsm-table> zu verwalten.*

1. Hier gibt es weitere Events, die auf den gleichen Zustand zeigen, z.B. Bewegung oder Verschieben (lt. Aufgabenstellung). Es werden hier zur Vereinfachung und zur Kompakthaltung der States und Events etwaige Nebeneffekte (z.B. Verschieben oder Graben) mit der Bewegung an sich zusammengefasst. Das v.a., da die Aufgabenstellung meiner Meinung nach sehr inkonsistent formuliert ist. [↑](#footnote-ref-1)
2. s. Fußnote 1 [↑](#footnote-ref-2)
3. Dass sich Patchbot „in Sicht“ befindet, heißt hier, dass jedes Tile t mit x und y zwischen Patchbot pb mit x und y und Gegener g mit x und y: nicht vom Typ Wand ist. Typ Wand sei hier terrain::CONCRETE\_WALL, terrain::ROCK\_WALL, terrain::AUTOMATIC\_DOOR, terrain::MANUAL\_DOOR oder terrain::SERET\_PASSAGE. Wohl gemerkt sind die Türen geschlossen. Wenn das Ergebnis der o.g. Funktionen direkt auf ein Tile fällt, d.h. , je nachdem ob , werden auch die diagonale nächsten Tiles mit geprüft (also x-1, y und x, y-1). [↑](#footnote-ref-3)