

## 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<sup>\*1)</sup> = Zielfeld nicht Wand?

	zb_u_ns	zf	zb	zf_u_ns	zs	zw	znw
→ FW*	W	–	–	FW	FaW	–	–
W	–	FW	W	–	–	–	–
FaW <sup>*)</sup>	–	–	–	–	–	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<sup>1</sup>

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	–	–	–	–

## DIGGER<sup>1</sup>

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
--	----	----------	----------	----

<sup>1</sup> 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.

<b>→ HB</b>	HB	VB	–	Z
<b>VB</b>	VB	–	HB	Z
<b>Z</b>	–	–	–	–

## SWIMMER<sup>2</sup>

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
<b>→ HB</b>	HB	VB	–	Z
<b>VB</b>	VB	–	HB	Z
<b>Z</b>	–	–	–	–

## FOLLOWER<sup>3</sup>

V = Verfolge  
 W = Warte

pb\_is = Patchbot in Sicht und erreichbar?  
 pb\_as = Patchbot außer Sicht oder nicht erreichbar?

	pb_is	pb_as
<b>→ W</b>	V	W
<b>V</b>	V	W

## HUNTER<sup>3</sup>

V = Verfolge  
 W = Warte  
 J = Jage

pb\_is = Patchbot in Sicht und erreichbar?  
 pb\_as = Patchbot außer Sicht oder nicht erreichbar?

<sup>2</sup> s. Fußnote 1

<sup>3</sup> Dass sich Patchbot „in Sicht“ befindet, heißt hier, dass jedes Tile t mit x und y zwischen Patchbot pb mit x und y und Gegner g mit x und y:  $x_t: x_{pb} \leq x_t \leq x_g \wedge y_t: y_t = \frac{y_g - y_{pb}}{x_g - x_{pb}} x_t + y_g - \frac{y_g - y_{pb}}{x_g - x_{pb}} x_g \vee x_t: x_t = \frac{(y_t - y_g - \frac{y_g - y_{pb}}{x_g - x_{pb}} x_g)}{\frac{y_g - y_{pb}}{x_g - x_{pb}}} \wedge y_t: y_{pb} \leq y_t \leq y_g$  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.  $y_t = \lfloor y_t \rfloor \vee x_t = \lfloor x_t \rfloor$ , je nachdem ob  $\Delta x > \Delta y$ , werden auch die diagonale nächsten Tiles mit geprüft (also x-1, y und x, y-1).

	<b>pb_is</b>	<b>pb_as</b>
<b>→ w</b>	V	-
<b>v</b>	V	J
<b>j</b>	V	J

---

# SNIFFER

V
= Verfolge

W
= Warte

pe
= Patchbot erreichbar?

pne
= Patchbot nicht erreichbar?

	<b>pe</b>	<b>pne</b>
<b>→ w</b>	V	W
<b>v</b>	V	W