

Practical UML Statecharts in C/C++, Second Edition: Event-Driven Programming for Embedded Systems

By Miro Samek

Newnes, 2008

ISBN-10: 0750687061

ISBN-13: 978-0750687065

Companion Website: www.state-machine.com/psicc2

Errata for first printing (October 2008)

1.1 Errors on the Back Cover:

Location	Is	Should be
First Bullet, "Understand State Machine Concepts"	From traditional finite state automated to modern UML state machines	From traditional finite state automata to modern UML state machines

1.2 Errors in the Text:

Location	Is	Should be
Page 15 Listing 1.1 explanation section (17)	(17) The function <code>QF_poolInit()</code> initializes...	(17) The function <code>QF_psInit()</code> initializes...
Page 20, Figure 1.4 explanation section (12)	... Missile posts the <code>MISSILE_IMG(x, y, bmp)</code> event to Table.	... Missile posts the <code>MISSILE_IMG(x, y, bmp)</code> event to Tunnel.
Page 20, Figure 1.4 explanation section (13)	Table renders the Missile bitmap...	Tunnel renders the Missile bitmap...
Page 26, Figure 1.6 explanation section (7)	The <code>PLAYER_TRIGGER</code> internal transition...	The <code>PLAYER_TRIGGER</code> internal transition...
Page 26, Figure 1.6 explanation section (8)	... The score is not posted to the Table at this point.	... The score is not posted to the Tunnel at this point.
Page 30, Figure 1.9 explanation section (7)	The exit action in the “used” state posts the <code>MINE_DISABLED(mine_id)</code> event to the Tunnel active object... Note that generating the <code>MINE_DISABLED(mine_id)</code> event in the exit section from “used” ...	The exit action in the “used” state posts the <code>MINE_DISABLED(mine_id)</code> event to the Tunnel active object... Note that generating the <code>MINE_DISABLED(mine_id)</code> event in the exit section from “used” ...
Page 31, Fig 1.9	The internal transition <code>TIME_TICK</code> in state “used” is: <pre>TIME_TICK [me->x + GAME_MISSILE_SPEED_X < GAME_SCREEN_WIDTH] / me->x += GAME_MISSILE_SPEED_X; postFIFO(Tunnel, MISSILE_IMG(me->x, me->y, MISSILE_BMP));</pre>	The internal transition <code>TIME_TICK</code> in state “used” should be: <pre>TIME_TICK [me->x >= GAME_SPEED_X] / me->x -= GAME_SPEED_X; postFIFO(Tunnel, MISSILE_IMG(me->x, me->y, MINE2_BMP));</pre> (see also state diagram below)

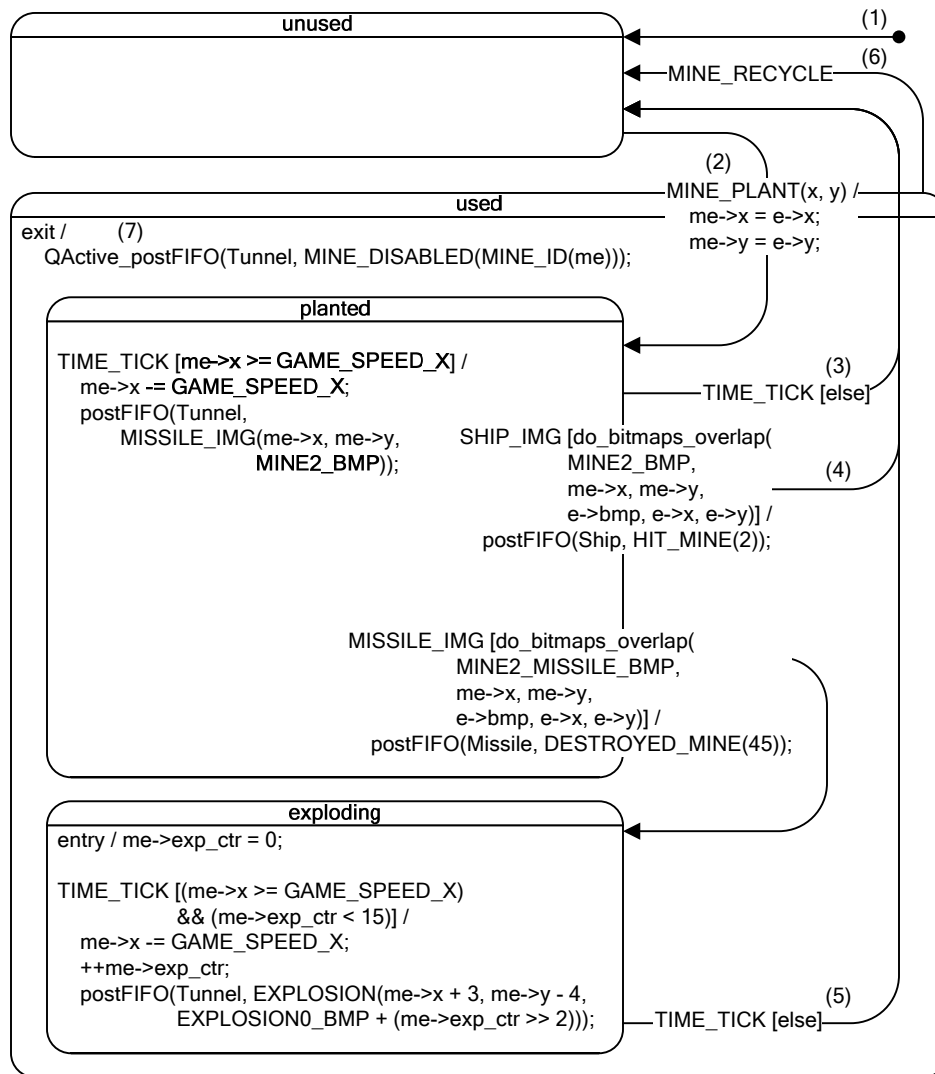


Figure 1.9: Mine2 state machine diagram.

Location	Is	Should be
Page 53, 4 th paragraph	While the coding ...	While the coding ...
Page 71, end of first paragraph	superstate "hating"	superstate "heating"
Page 88, Fig 2.11	The transition G in state "s21" goes to "s11"	The transition G in state "s21" should go to "s1" (see also state diagram below)

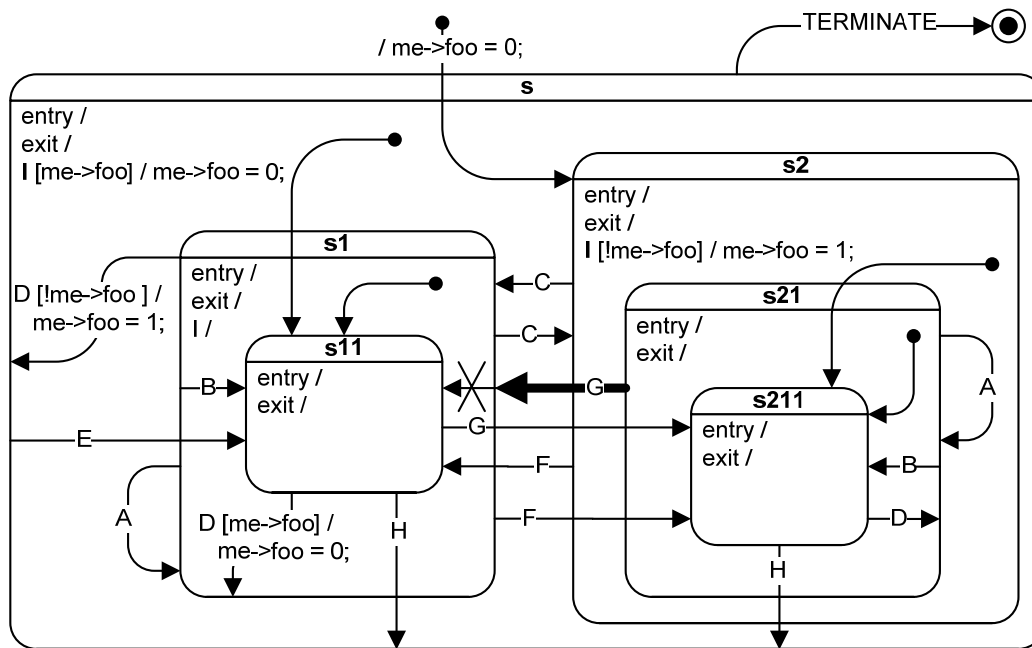


Figure 2.11: Hypothetical state machine that contains all possible state transition topologies up to four levels of state nesting.

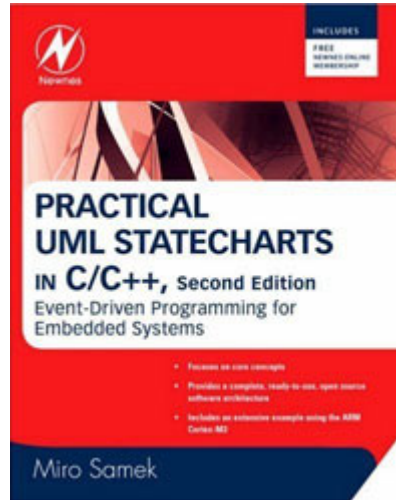
Location	Is	Should be
Page 96, Section 2.4.6, end of paragraph	The actual C implementation...	The actual C++ implementation...
Page 116, Listing 3.2 explanation section (3)	This typedef defines Tran type as a pointer to the StateTable struct and a pointer to the Event struct as arguments and returns uint8_t. The value returned from the transition function represents the next state for the state machine after executing the transition...	This typedef defines Tran type as a pointer to the StateTable struct and a pointer to the Event struct as arguments and returns void....
Page 121, Listing 3.4 explanation section (17)	The sate table ...	The state table ...
Page 137, Listing 3.7(7)	<pre>(void) (*me->state)(me, &QEP_reservedEvt_[Q_EXIT_SIG]);</pre>	<pre>(void) (*me->state)(me, &QEP_reservedEvt_[Q_ENTRY_SIG]);</pre>
Page 156, footnote no. 4	Casting from subclass to superclass is called in OOP downcasting...	Casting from superclass to subclass is called in OOP downcasting...
Page 160, Listing 4.4(6)	<code>Q_HANLDED()</code>	<code>Q_HANDLED()</code>
Page 170, Section 4.5.6 Bullet 4.	Execution of the entry actions to the "result" state	Execution of the entry actions to the "ready" state
Page 170, Section 4.5.6	Execution of the actions associated with the initial transition defined in	Execution of the actions associated with the initial transition defined in the "ready"

Bullet 5.	the “result” state	state
Page 210, Listing 5.1 explanation section (1)	Every QEP application needs to include <code>qep_porth.h</code>	Every QEP application needs to include <code>qep_port.h</code>
Page 247, Listing 5.9 Comment line	HSM definitio-----	HSM definition-----
Page 295, 2 nd paragraph from the bottom	... code to “ wonder around,” silently taking care of code to “ wander around,” silently taking care of ...
Page 398 1 st paragraph	The header file <code>qep_porth.h</code> ...However, <code>qep_porth.h</code> also...	The header file <code>qep_port.h</code> ...However, <code>qep_port.h</code> also...
Page 413 1 st paragraph in Section 8.1.9,	As you design you port, you must decide...	As you design your port, you must decide...
Page 569, Section 11.3.7, 2 nd paragraph	You can employ just about any repetition physical data link available...	You can employ just about any physical data link available...

1.3 Contact Information

Quantum Leaps, LLC
103 Cobble Ridge Drive
Chapel Hill, NC 27516
USA

+1 866 450 LEAP (toll free, USA only)
+1 919 869-2998 (FAX)
e-mail: info@quantum-leaps.com
WEB : <http://www.quantum-leaps.com>
<http://www.state-machine.com>



“Practical UML Statecharts in C/C++, Second Edition: Event Driven Programming for Embedded Systems”, by Miro Samek, Newnes, 2008

