## A simplified schema of action-processing rules and rulebooks

									Rulebook	if rulebook ends	ACTION	
Rulebook	Rule	Routine	Routine	Rulebook	Rule	Routine	Rulebook	Rule	(specific bold)	(default in bold)	outcome ¤	(stage rule returns)
ΓSR	GenerateAction	BeginAction A	ctionPrimitive	APR		>	•	*Before stage	BEFORE	Rule succeeds	SUCCESS	true or false
			<rt f<="" td=""><td></td><td> </td><td></td><td></td><td></td><td></td><td>Rule fails</td><td>Failure</td><td></td></rt>							Rule fails	Failure	
			1							Make no decision	(SUCCESS)	
			1							instead/stop	±	
		(if meta (	out of world))				>	*Instead stage	INSTEAD	Rule succeeds	SUCCESS	true or false
			1							Rule fails	Failure	
			1							Make no decision	(SUCCESS)	
			1							instead/stop	±	
				>	DSAPR>	#actions_table>(action+1)	SAPR>	§ Check stage	CHECK	Rule succeeds	SUCCESS	2 or false (2 is true)
			< r	true						Rule fails	Failure	
										Make no decision	(SUCCESS)	
										instead/stop	±	
							>	`Carry out stage	CARRY OUT	Rule succeeds	(SUCCESS)	false => no outcome
	The specific action									Rule fails	(SUCCESS)	
	(the specific check									Make no decision	(SUCCESS)	
	and specific report	•								instead/stop	± (SUCCESS)	
	set at the start of t	•	n rulebook				>	*After stage	AFTER	Rule succeeds	SUCCESS	true or false
	according to the a						1			Rule fails	Failure	
	e.g. for a Take acti	on, the Check tal	king rulebook etc				1			Make no decision	(SUCCESS)	
							1			instead/stop	±	
							>	`Report stage	REPORT	Rule succeeds	(SUCCESS)	false => no outcome
	TSR - turn sequence						1			Rule fails	(SUCCESS)	
	APR - action-proce	-					1			Make no decision	(SUCCESS)	
	DSAPR - descend t						1			instead/stop	± (SUCCESS)	
	SAPR - specific act	ion-processing ru	iles				>	Action succeeds*	*		SUCCESS	true => ends rulebook in success

<sup>\* (</sup>abides by the associated rule book)

#actions\_table is a compiler-generated constant pointer to the game file's action table- also a creation of the compiler

The specific entry for an action calls the relevant VerbSub(), which after setting action parameters for meta, keep\_silent, and the action-specific rulebooks for Check <action>, Carry out <action>, Report <action>, goes on to call the SpecificActionProcessingRulebook.

bracketed results (SUCCESS) indicate the outcome of the action if no subsequent rulebook intervenes by making a decision (by returning true) Note that BEFORE, INSTEAD, CHECK, AFTER stage rules can decide the action, but not CARRY OUT or REPORT (which both always return false)

<sup>§ (</sup>anonymously abides by the associated rulebook)

<sup>`(</sup>follows the associated rule book)

x bracketed outcomes are those which will ensue if no subsequent rule makes a decision

<sup>±</sup> decide this rulebook, with success/failure/no outcome as decided by the rule/action just called, or by the last preceding rule e.g. try looking instead / follow the inedible item rules instead. / try looking; stop the action.

NB follow <rules ending in no outcome> instead will, unlike `make no decision', immediately end APR and the action with no outcome

(This means, in contrast to an explicit 'make no decision' ending, that the action, not having succeeded, will subsequently return false to 'If we have...')

<sup>\*\*</sup> this rule ensures that if no other rulebook makes a decision, (or stops the rulebook with no outcome) SAPR succeeds by default and this result feeds back via DSAPR +/- APR to ActionPrimitive(), so that the action is logged as having succeeded at least once and 'if we have...' becomes true.