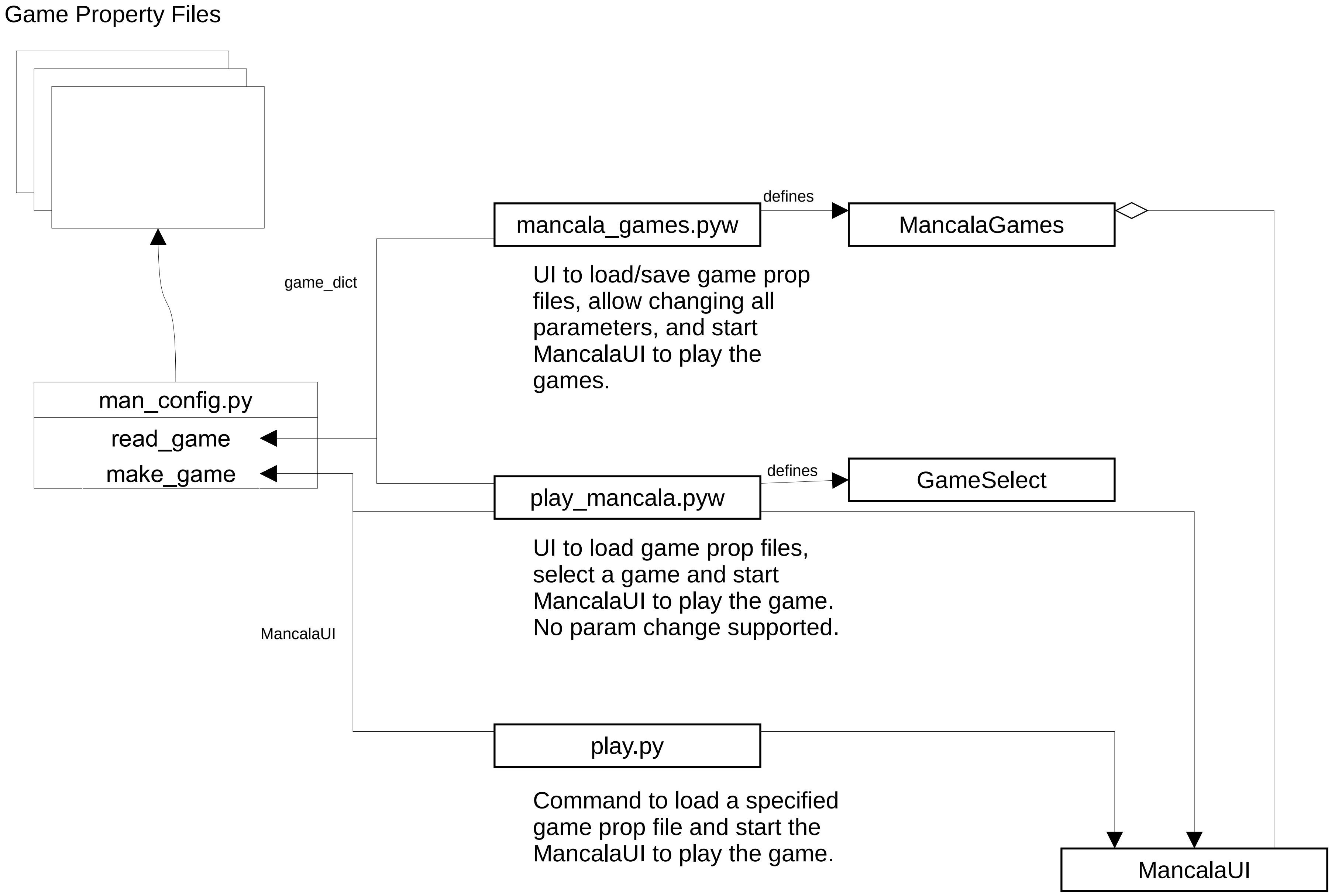
Mancala Games



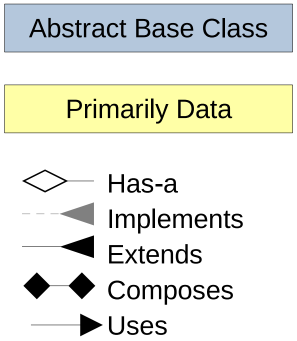
UI to load/save game prop files, allow changing all parameters, and start MancalaUI to play the games.

UI to load game prop files, select a game and start MancalaUI to play the game. No param change supported.

Command to load a specified game prop file and start the MancalaUI to play the game.

Notation Convensions

Class Diagram Convensions



Abstract Base Class

Primarily Data

Has-a

Implements

Extends

Composes

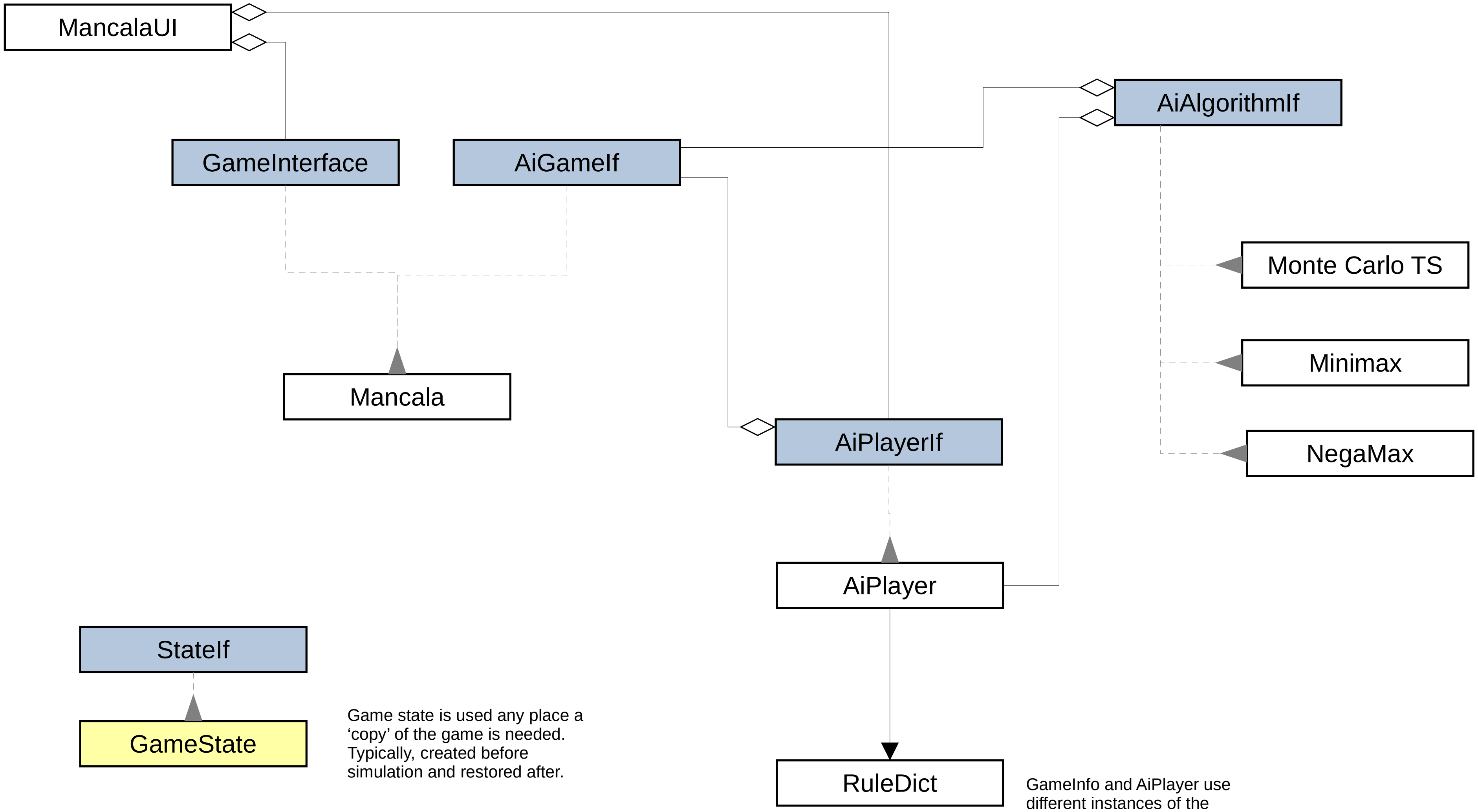
Uses

Deco Chain Convensions

- One path down the deco chain is used.
- Intersecting arrows are decision points.
- Shown in **call order** from start dot (constructed in reverse order). Calls down the deco chain maybe at any point in a deco’s processing.
- All paths shown might not be possible (see ginfo_rules).

🟥 Optional deco

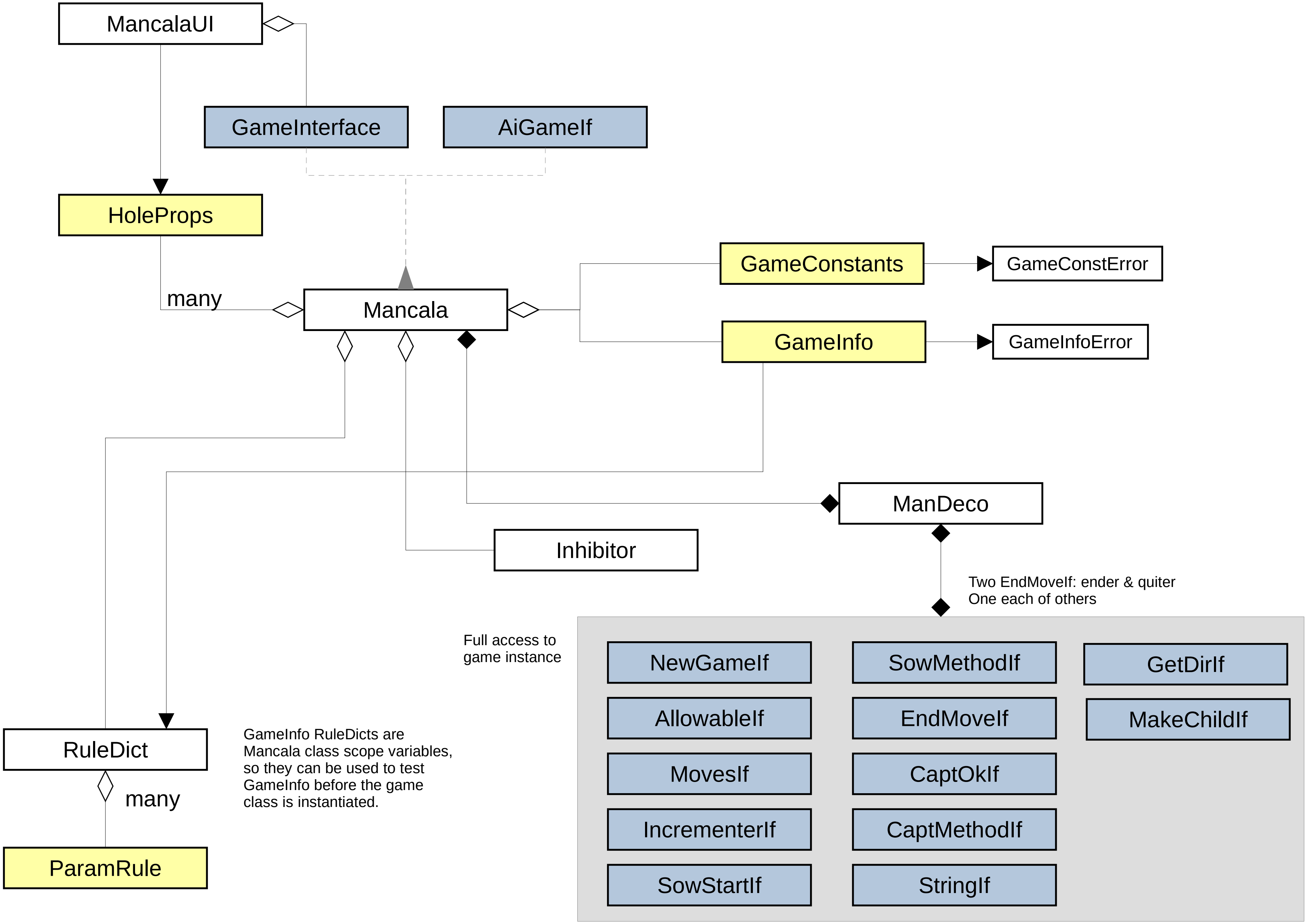
Mancala UI, Mancala & AI Classes



Game state is used any place a ‘copy’ of the game is needed. Typically, created before simulation and restored after.

GameInfo and AiPlayer use different instances of the RuleDict

Mancala Classes



MancalaUI

GameInterface

AiGameIf

HoleProps

many

Mancala

GameConstants

GameConstError

GameInfo

GameInfoError

ManDeco

Inhibitor

Two EndMoveIf: ender & quiter
One each of others

Full access to
game instance

NewGameIf

SowMethodIf

GetDirIf

AllowableIf

EndMoveIf

MakeChildIf

MovesIf

CaptOkIf

IncrementerIf

CaptMethodIf

SowStartIf

StringIf

RuleDict

many

ParamRule

GameInfo RuleDicts are
Mancala class scope variables,
so they can be used to test
GameInfo before the game
class is instantiated.

Decorator Usage

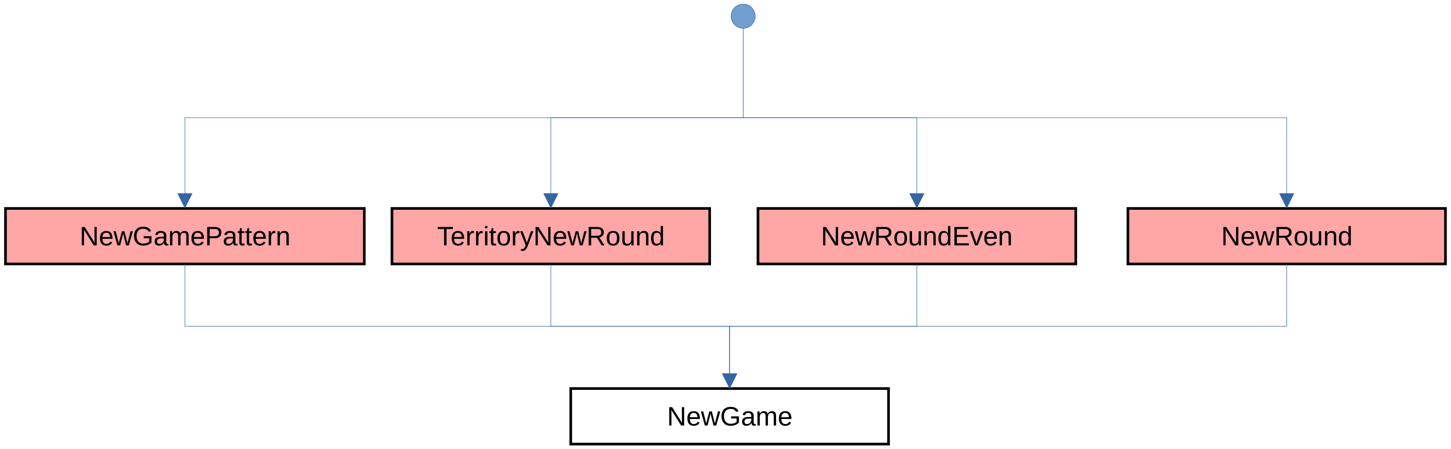
Mancala Move Steps

Mancala Methods (mancala.py)		Description	Decorator(s)
move	do_sow	Start Sow (parse move, first hole, seeds)	sow_start
		Get Direction (CW or CCW)	get_direction
		Sow – drop seeds	sower, capt_ok, incrementer & make_child
	capture_seeds	Capture Seeds	capt_ok, capturer, incrementer & make_child
	win_conditions	Win Condition – is game over	ender

Decorator Calls (non-move)

Interface	Method (mancala.py)	Decorator
GameInterface	new_game	new_game
GameInterface	end_game	quitter
GameInterface	get_allowable_holes	allow
AIGameIf	get_moves	get_moves
not applicable	__str__	get_string

New Game Decorators and Chain



NewGamePattern

TerritoryNewRound

NewRoundEven

NewRound

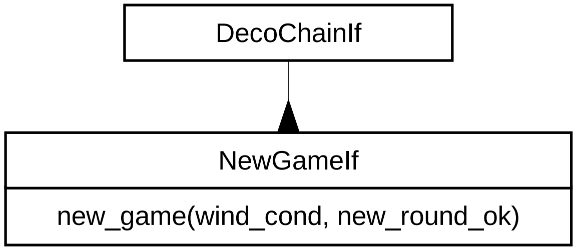
NewGame

State variables changed:

- blocked
- board
- owner
- starter
- store
- turn

Parameters:

- blocks
- goal
- min_move
- round_starter
- round_fill
- rounds
- start_pattern

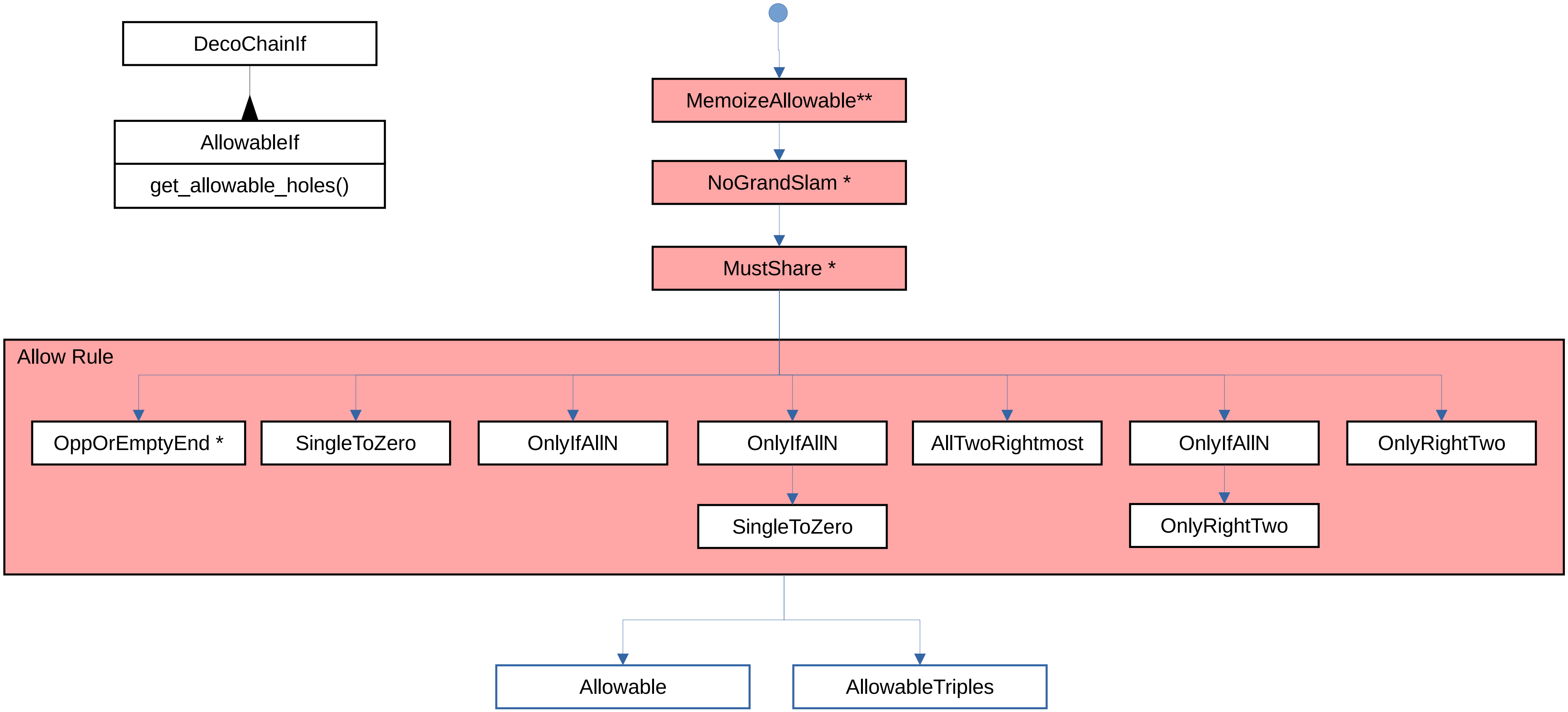


DecoChainIf

NewGameIf

new_game(wind_cond, new_round_ok)

Allowables Decorators and Chain



DecoChainIf

AllowableIf

get_allowable_holes()

MemoizeAllowable**

NoGrandSlam *

MustShare *

Allow Rule

OppOrEmptyEnd *

SingleToZero

OnlyIfAllN

OnlyIfAllN

SingleToZero

AllTwoRightmost

OnlyIfAllN

OnlyRightTwo

OnlyRightTwo

Allowable

AllowableTriples

State variables read:

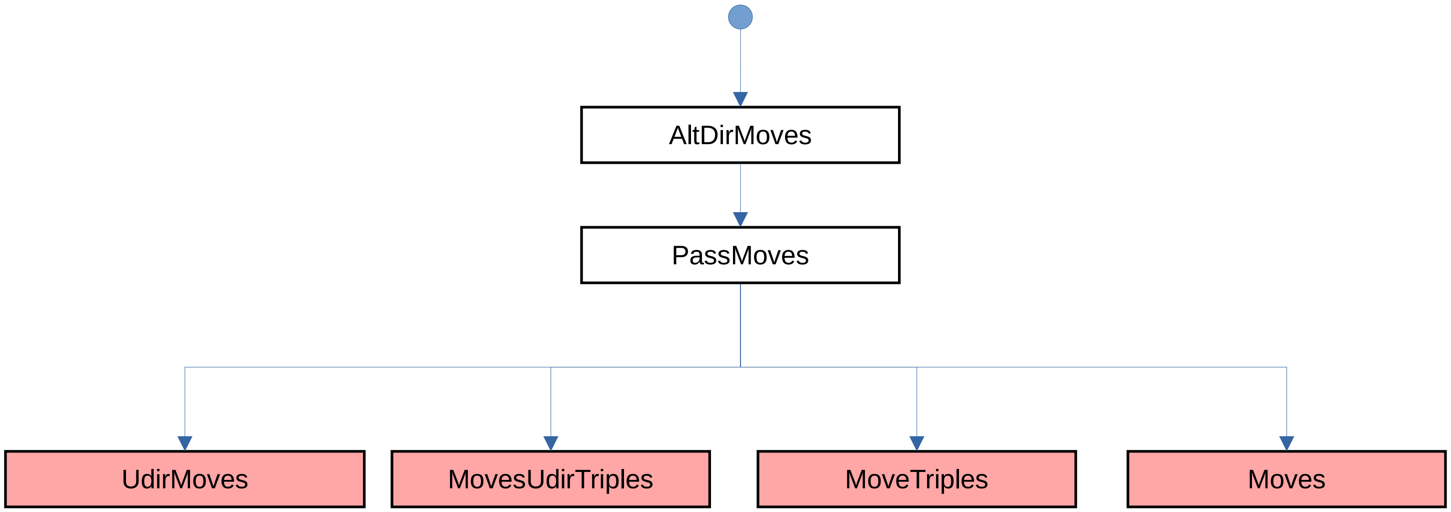
- turn
- board
- store
- blocked
- owner
- child
- mcount

Parameters:

- min_move
- allow_rule
- mlength
- mustshare
- grandslam

Notes:
Some allow rule decos are shown more than once for clarity.
* Simulates some portion of moves to determine allowables
** MemoizeAllowable is used for deco’s that simulate moves

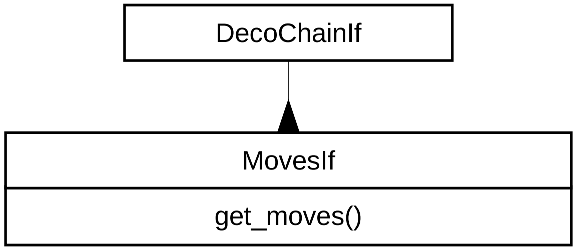
Get Moves Decorators and Chain



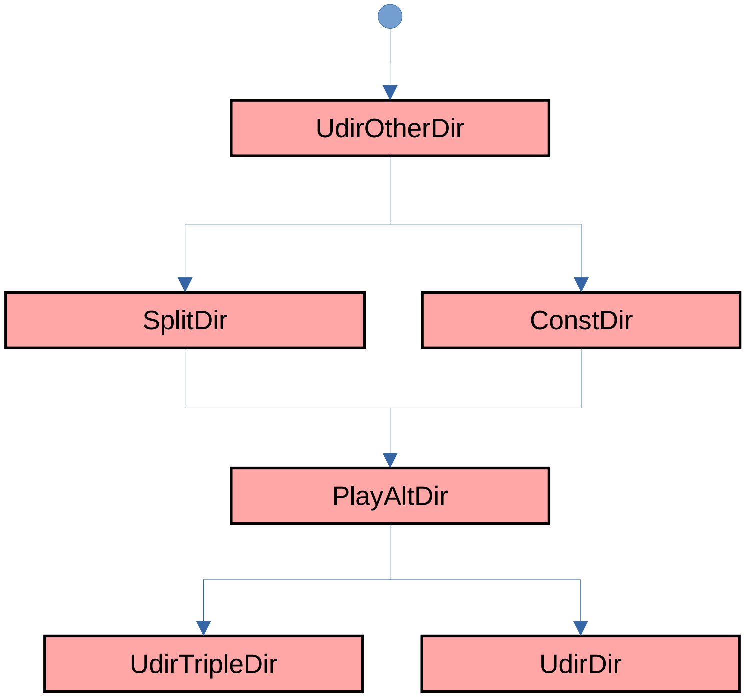
One of these
is required

State variables read:
- blocked
- board
- owner
- starter
- store
- turn

Parameters:
- mlength
- mustpass
- sow_direct
- udir_holes
- udirect



Get Direction Decorators and Chain



UdirOtherDir

SplitDir

ConstDir

PlayAltDir

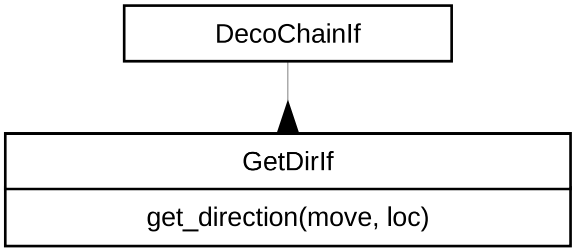
UdirTripleDir

UdirDir

One of these
is required

State variables read:
- mcount
- turn

Parameters:
- no_sides
- sow_direct
- udir_holes
- udirect

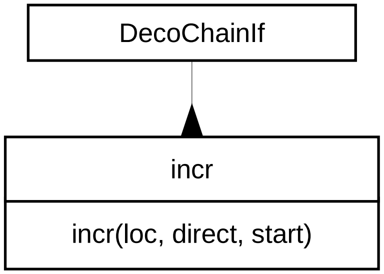


DecoChainIf

GetDirIf

get_direction(move, loc)

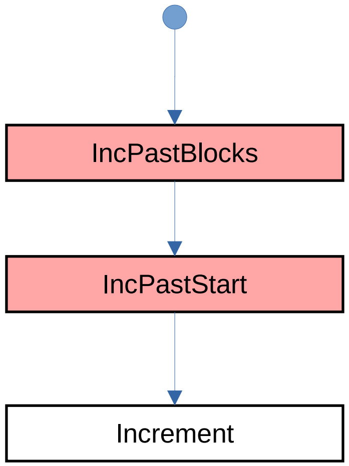
Incrementer Decorators and Chains



DecoChainIf

incr

incr(loc, direct, start)



IncPastBlocks

IncPastStart

Increment

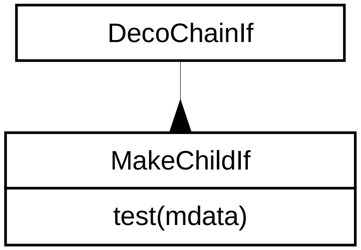
State variables read:
blocked

Parameters:
blocks
skip_start

MakeChild Decorator and Chain

State variables read:
board
child
inhibitor
owner
turn

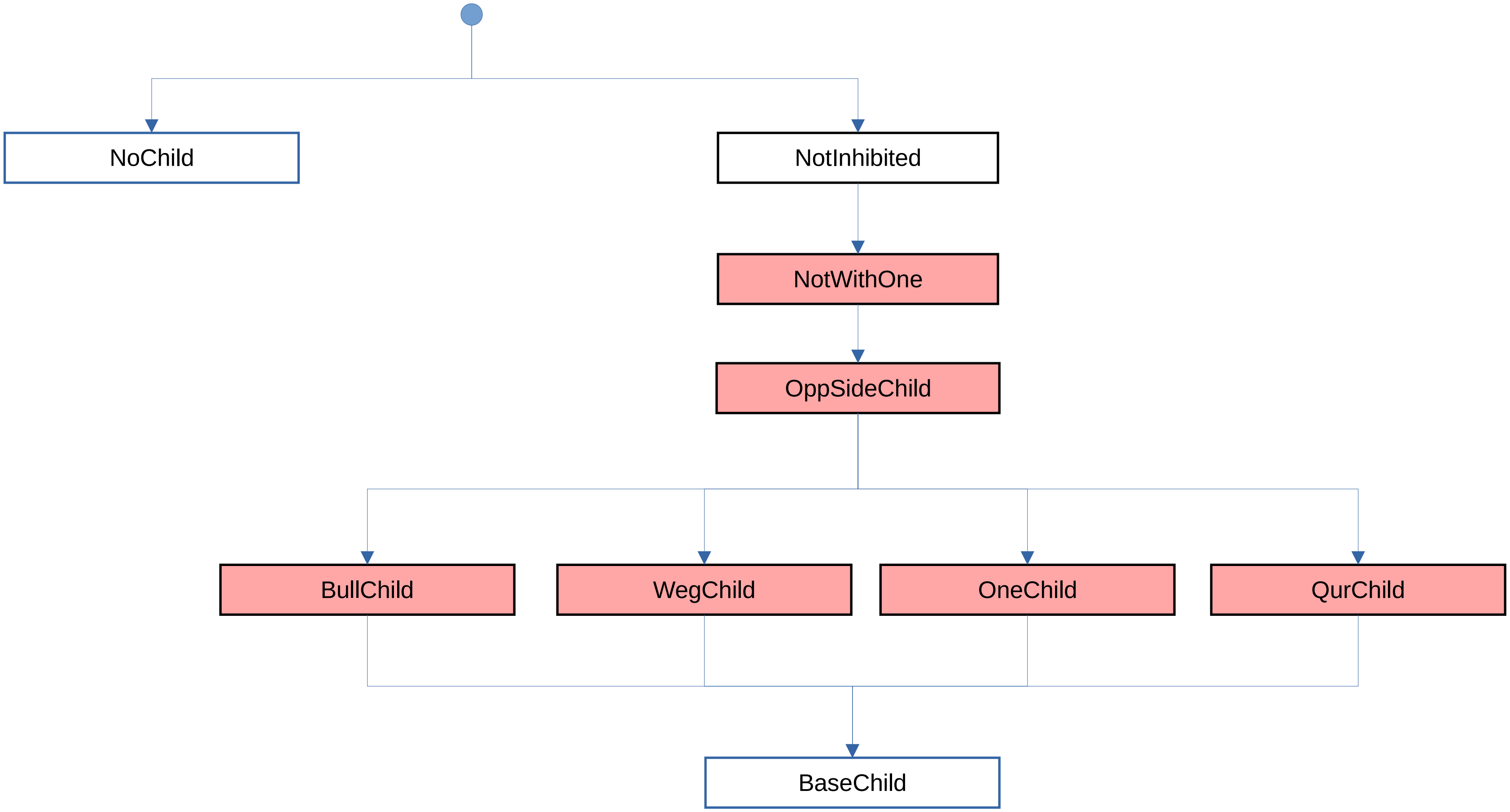
Parameters:
child_cvt
child_rule
child_type



DecoChainIf

MakeChildIf

test(mdata)



NoChild

NotInhibited

NotWithOne

OppSideChild

BullChild

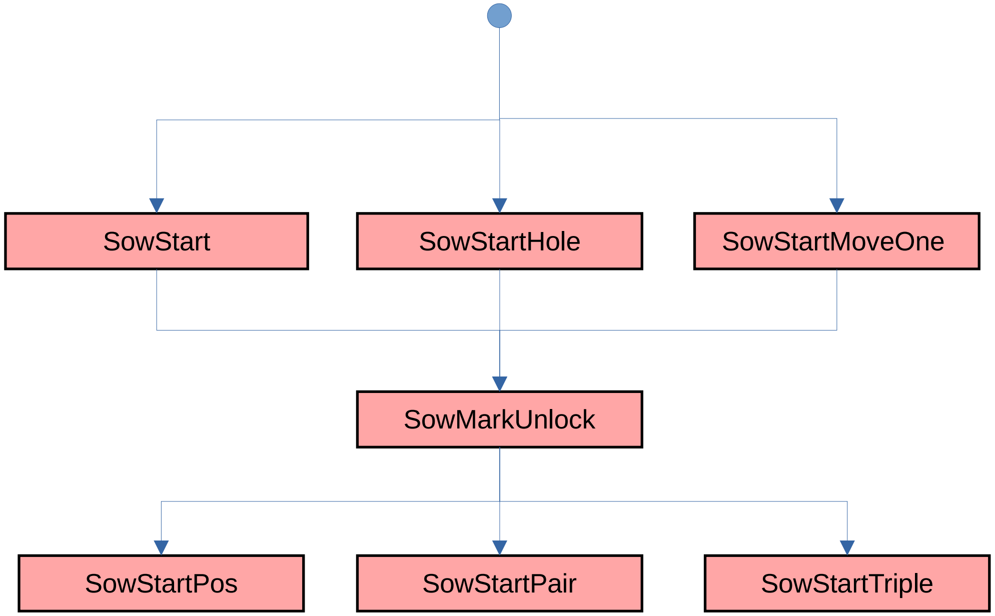
WegChild

OneChild

QurChild

BaseChild

Sow Starter Decorators and Chain



One of these is required

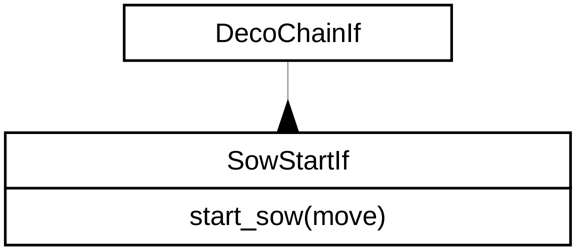
One of these is required

State variables:

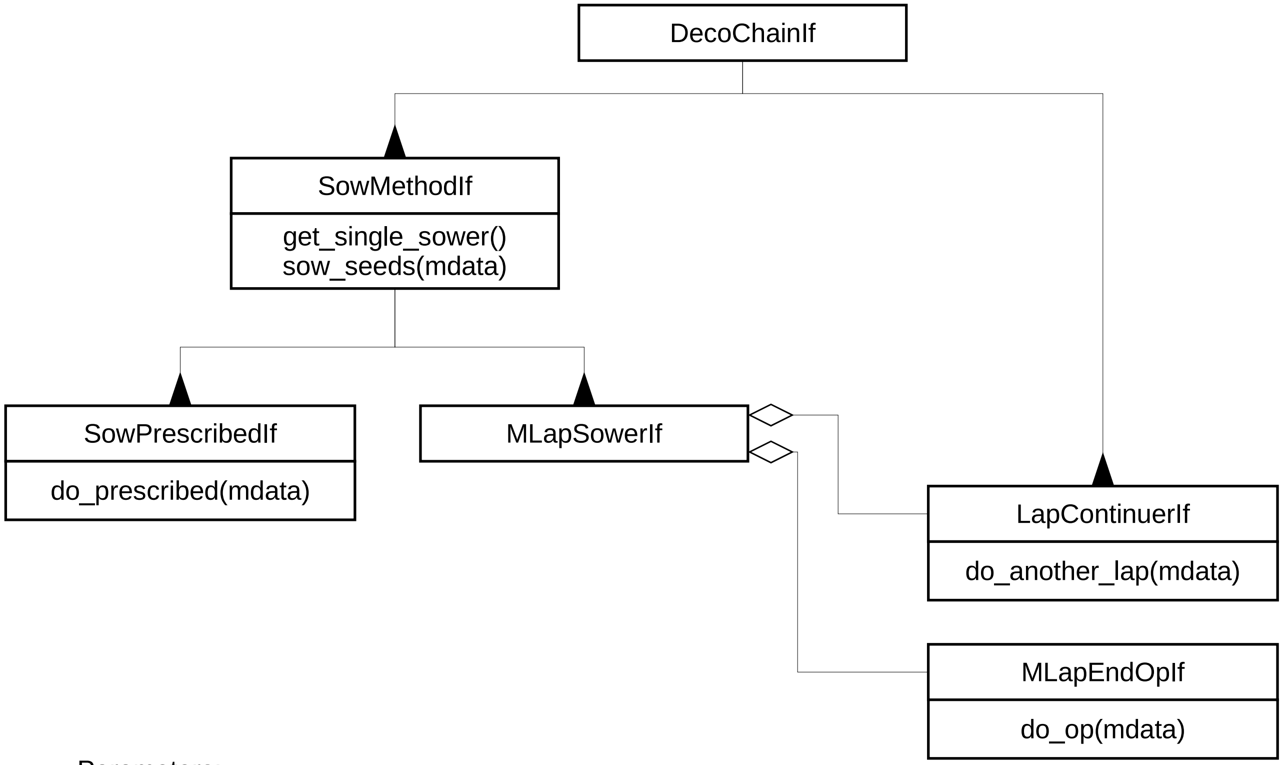
- Read:
  - turn
- Changed:
  - board
  - unlocked

Parameters:

- mlength
- move_one
- moveunlock
- sow_start



Sower Decorators



DecoChainIf

SowMethodIf

get_single_sower()
sow_seeds(mdata)

SowPrescribedIf

do_prescribed(mdata)

MLapSowerIf

LapContinuerIf

do_another_lap(mdata)

MLapEndOpIf

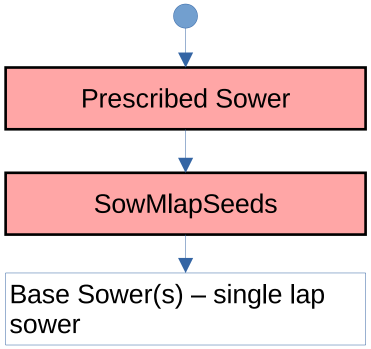
do_op(mdata)

State variables:

- Reads
  - inhibitor
  - turn
  - child
  - mcount
- Changes
  - board
  - store
  - blocked

Parameters:

- capt_max
- capt_min
- capt_on
- child_type
- crosscapt
- evens
- goal
- gparam_one
- mlaps
- prescribed
- sow_direct
- sow_own_store
- sow_rule
- visit_opp

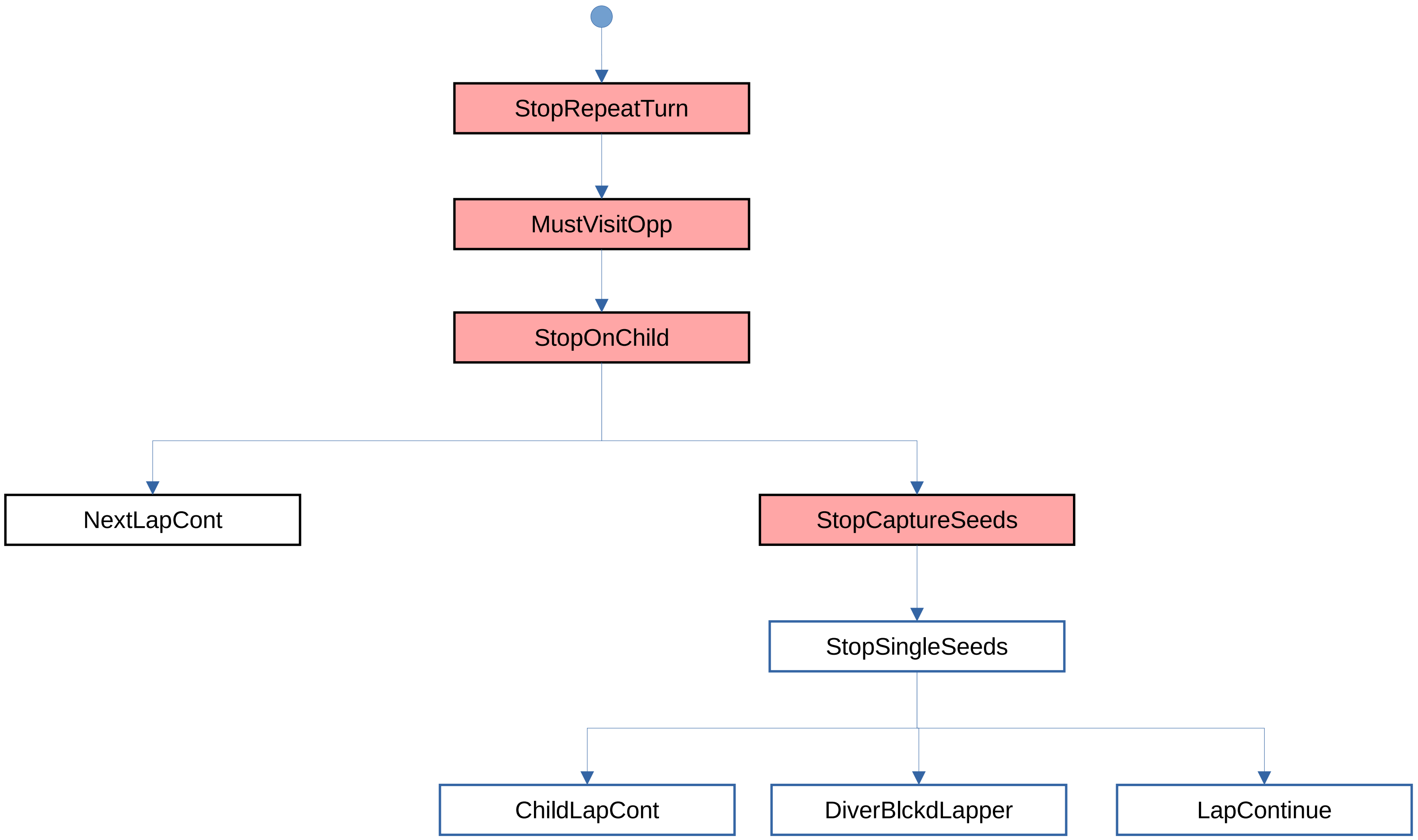


Prescribed Sower

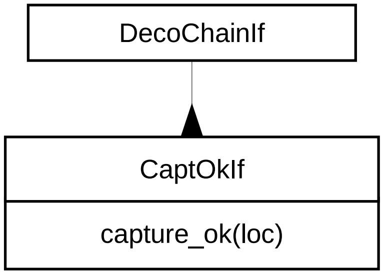
SowMlapSeeds

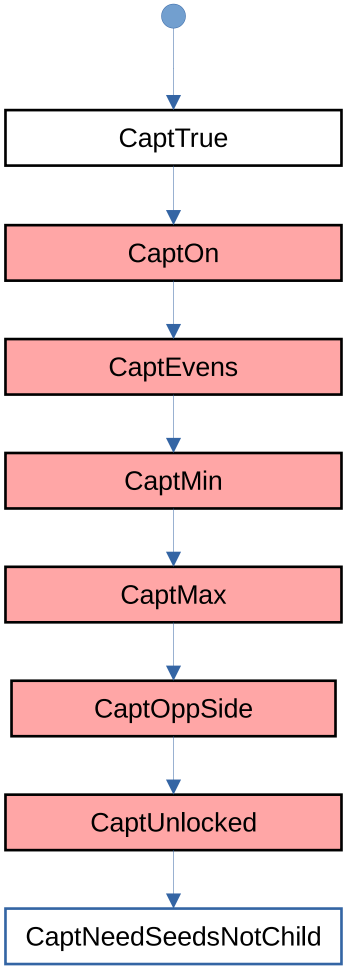
Base Sower(s) – single lap sower

Lapper Continuer Deco Chain



Capt Ok Decorators and Chains





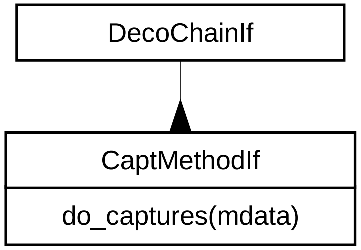
State variables read:
- board
- child
- turn
- unlocked

Parameters:
- capt_max
- capt_min
- capt_on
- moveunlock
- oppsidecapt

Captuer Decorators

State variables
- Reads
  - inhibitor
  - starter
  - turn
- Changes
  - board
  - child
  - store

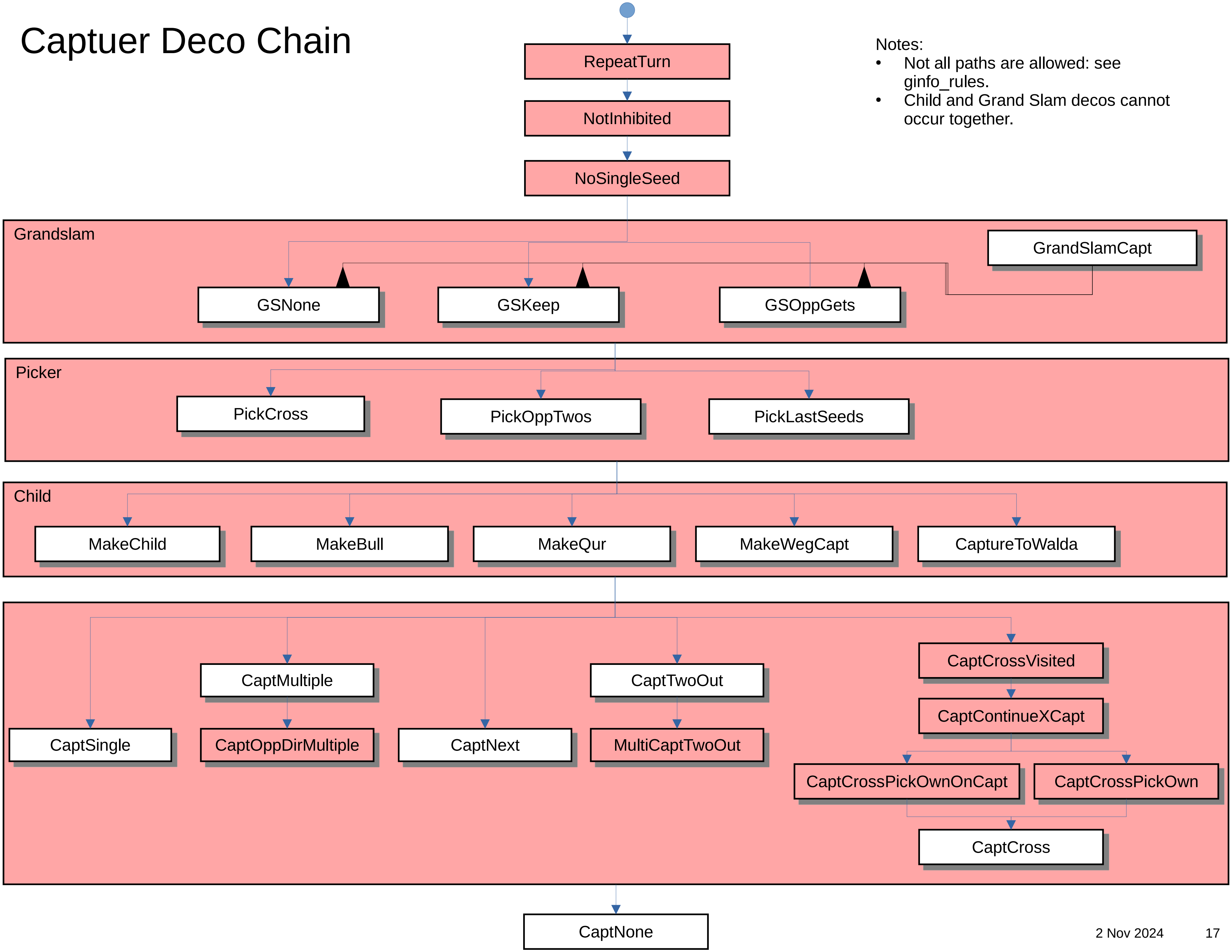
Parameters:
- capsamedir
- capt_max
- capt_min
- capt_next
- capt_on
- capt_rturn
- capttwoout
- child_cvt
- child_type
- crosscapt
- evens
- grandslam
- mlaps
- multicapt
- nocaptfirst
- nosinglecapt
- oppsidecapt
- pickextra
- prescribed
- round_fill
- xc_sown
- xcpickown



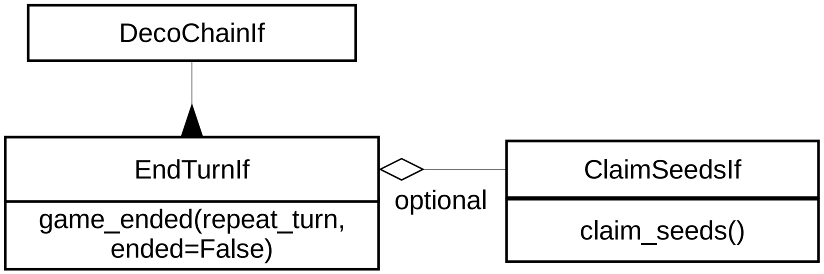
Captuer Deco Chain

Notes:

- Not all paths are allowed: see ginfo_rules.
- Child and Grand Slam decos cannot occur together.



Ender & Quiter Decorators and Chains



DecoChainIf

EndTurnIf

game_ended(repeat_turn, ended=False)

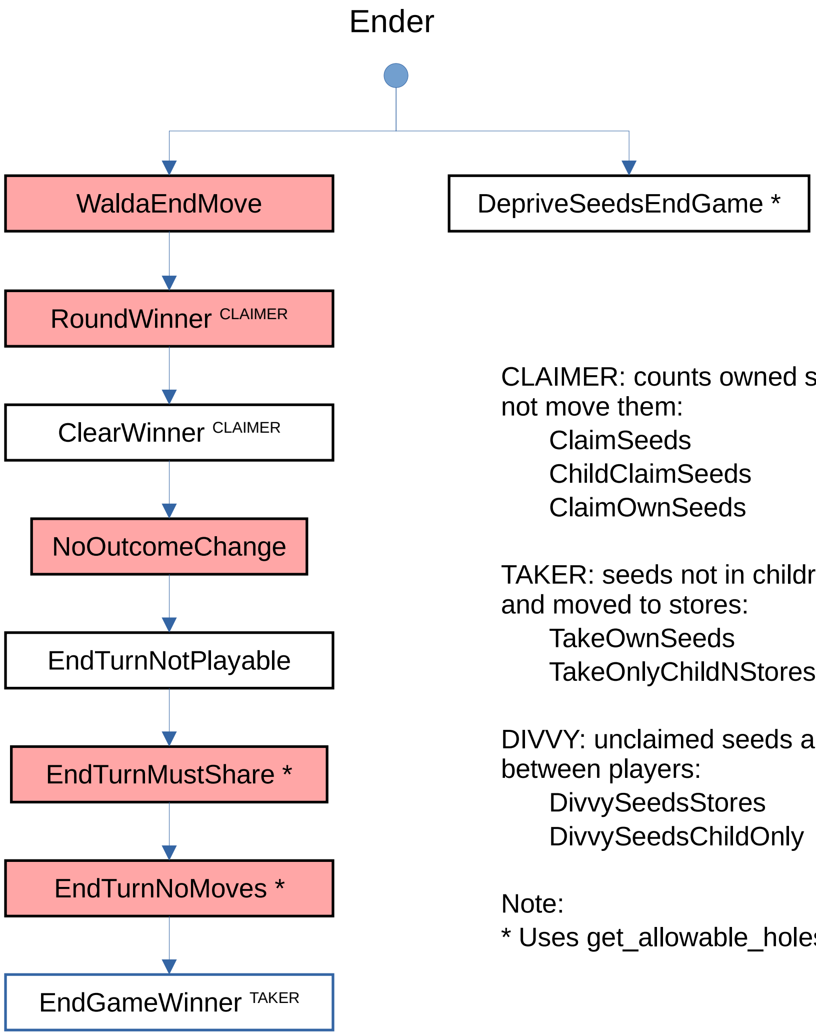
optional

ClaimSeedsIf

claim_seeds()

State variables:
- Reads:
  - child
  - owner
  - turn
- Changes:
  - board
  - store

Parameters:
- capt_min
- capt_next
- capt_on
- capttwoout
- child_cvt
- child_type
- crosscapt
- evens
- goal
- gparam_one
- min_move
- mlaps
- mustpass
- mustshare
- no_sides
- round_fill
- rounds
- sow_own_store
- stores



Ender

WaldaEndMove

DepriveSeedsEndGame *

RoundWinner CLAIMER

ClearWinner CLAIMER

NoOutcomeChange

EndTurnNotPlayable

EndTurnMustShare *

EndTurnNoMoves *

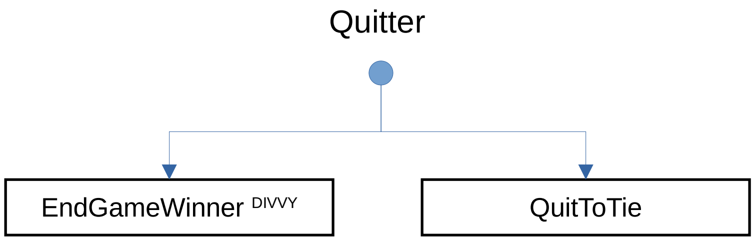
EndGameWinner TAKER

CLAIMER: counts owned seeds but does not move them:
- ClaimSeeds
- ChildClaimSeeds
- ClaimOwnSeeds

TAKER: seeds not in children are claimed and moved to stores:
- TakeOwnSeeds
- TakeOnlyChildNStores

DIVVY: unclaimed seeds are split between players:
- DivvySeedsStores
- DivvySeedsChildOnly

Note:
* Uses get_allowable_holes



Quitter

EndGameWinner DIVVY

QuitToTie