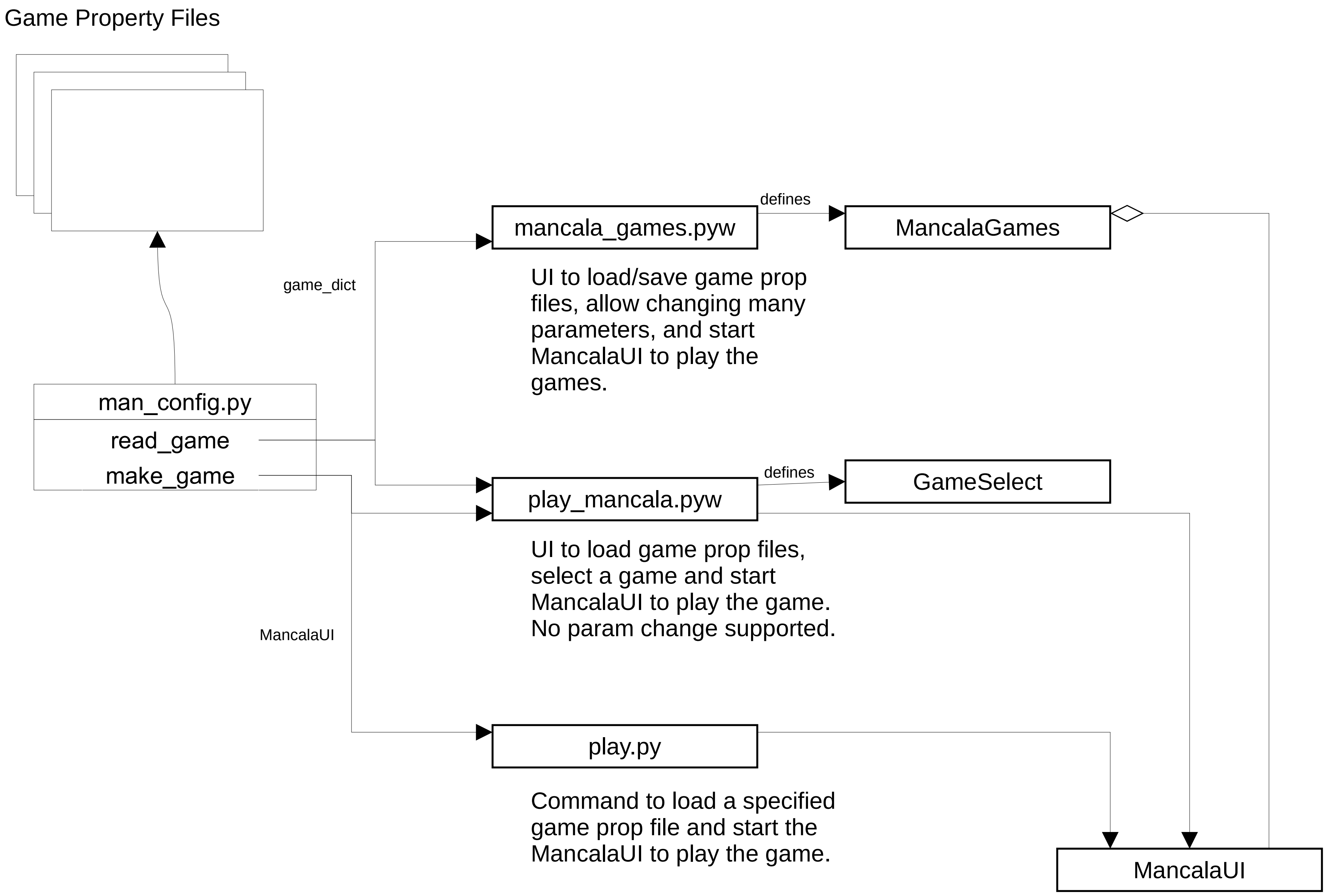
Mancala Games



Game Property Files

man_config.py

read_game

make_game

game_dict

MancalaUI

mancala_games.pyw

defines

MancalaGames

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

play_mancala.pyw

defines

GameSelect

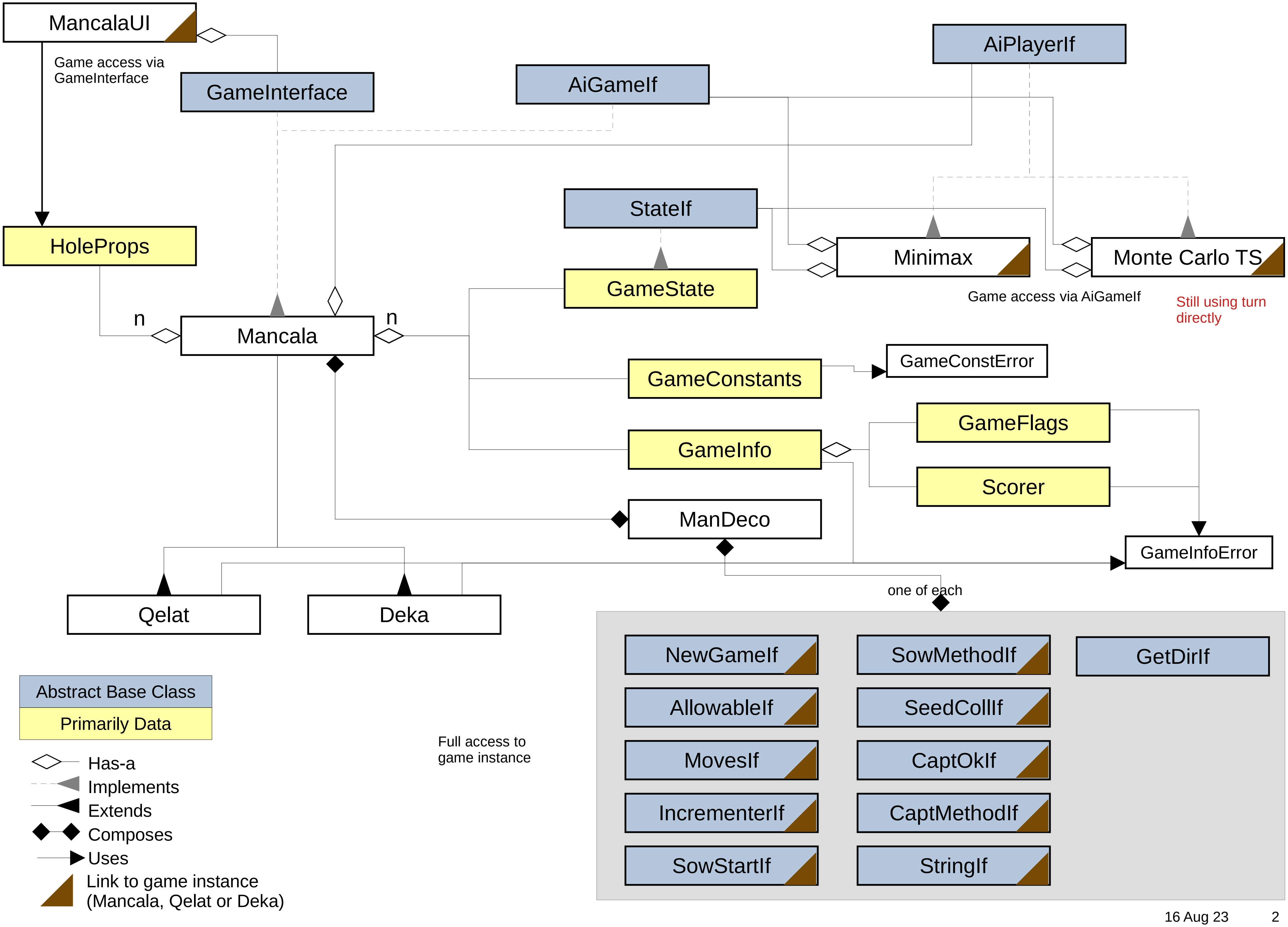
UI to load game prop files, select a game and start MancalaUI to play the game. No param change supported.

play.py

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

MancalaUI

Mancala Game Classes



MancalaUI

Game access via GameInterface

GameInterface

AiGameIf

AiPlayerIf

HoleProps

StateIf

Minimax

Monte Carlo TS

GameState

Game access via AiGameIf

Still using turn directly

n

Mancala

n

GameConstants

GameConstError

GameFlags

GameInfo

Scorer

ManDeco

GameInfoError

one of each

Qelat

Deka

NewGameIf

SowMethodIf

GetDirIf

AllowableIf

SeedCollIf

MovesIf

CaptOkIf

IncrementerIf

CaptMethodIf

SowStartIf

StringIf

Full access to game instance

Abstract Base Class

Primarily Data

Has-a

Implements

Extends

Composes

Uses

Link to game instance (Mancala, Qelat or Deka)

Decorator Usage

Mancala Move Steps

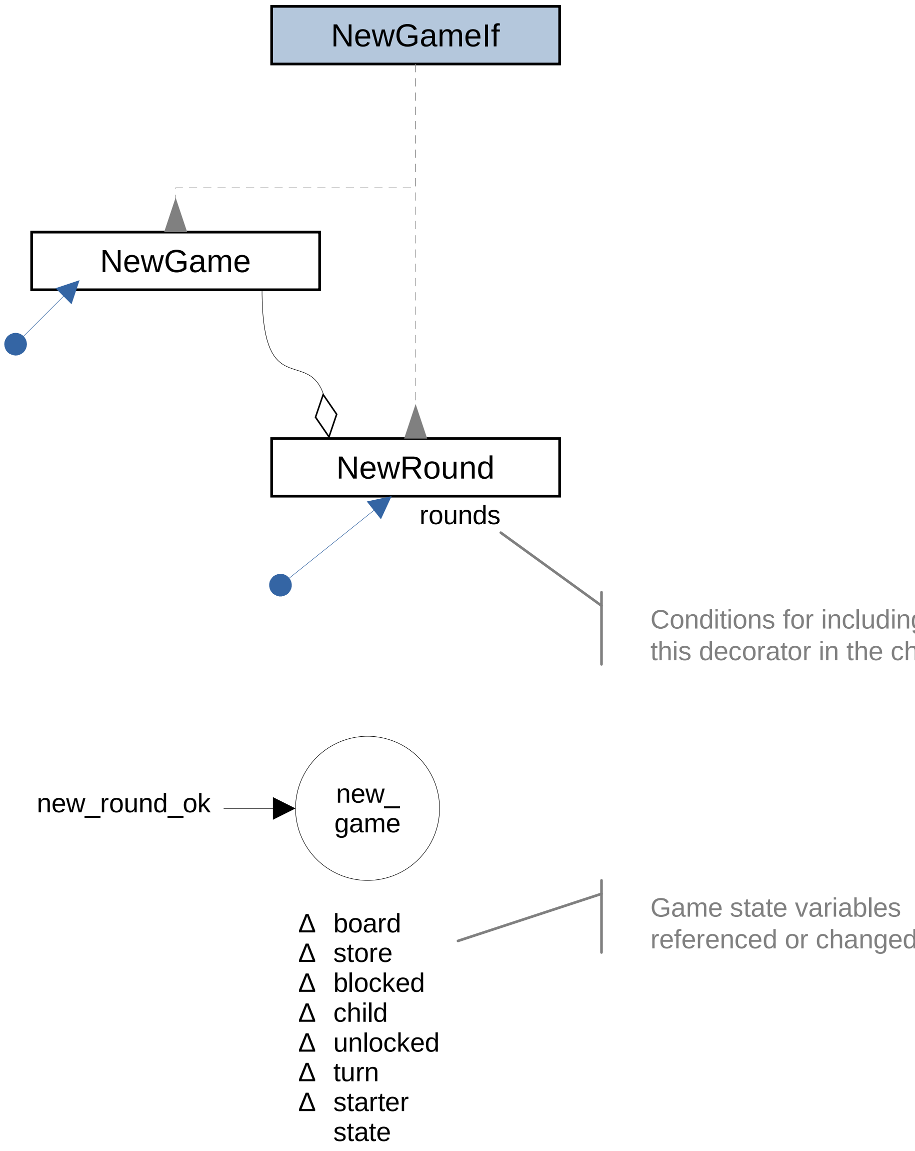
Mancala Methods (mancala.py)		Step Number	Description	Decorator(s)
move		1	Parse move & execute pass	
	do_sow	2	Start Sow (first hole, seeds)	sow_start
		3	Get Direction (CW or CCW)	get_direction
		4	Sow – drop seeds	sower & incr
	capture_seeds	5	Capture Seeds	capt_ok, capturer & incr
	win_conditions	6	Win Condition – is game over	collector

Decorator Calls (non-move)

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

New Game Classes and Deco

new_game.py

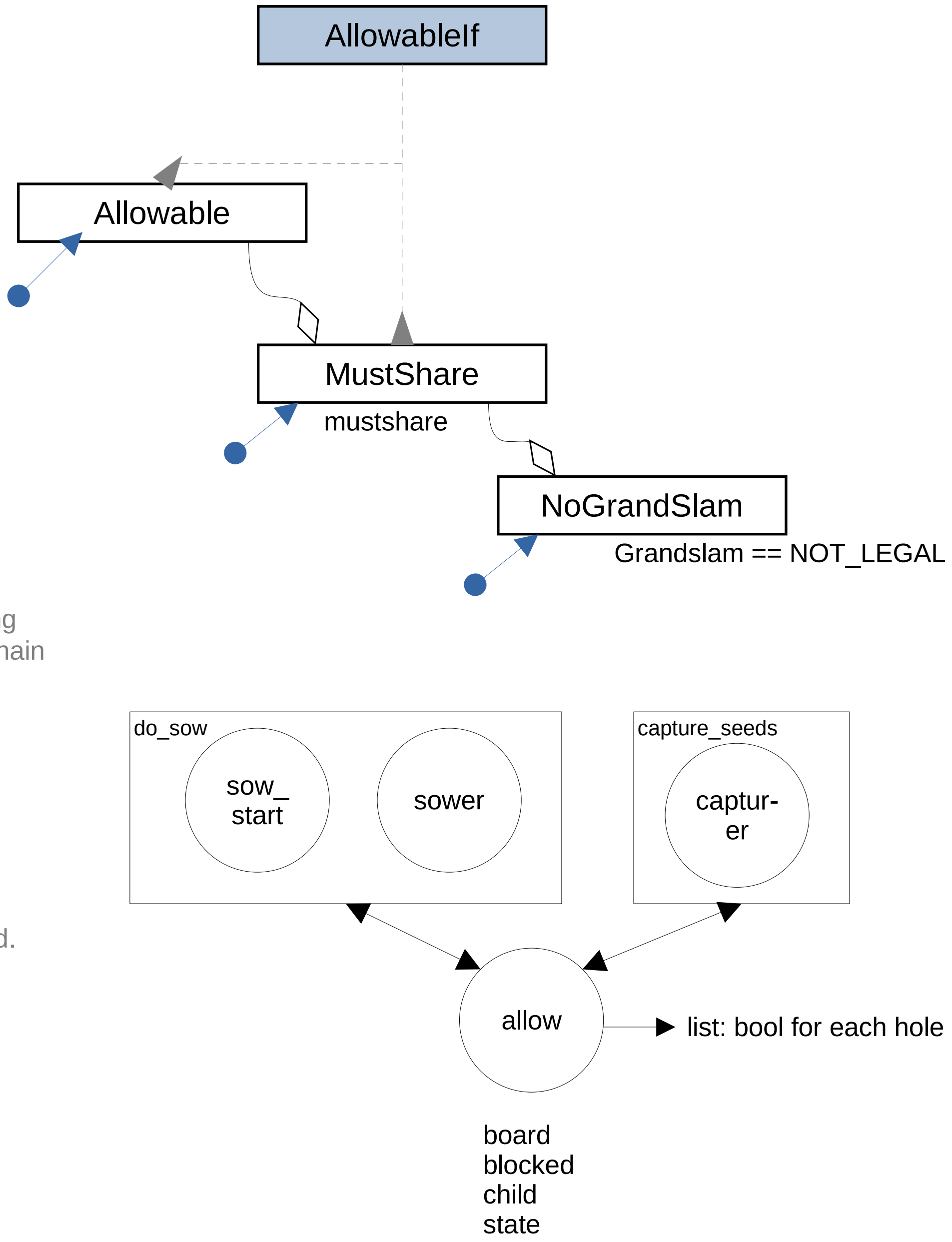


Conditions for including this decorator in the chain

Game state variables referenced or changed.

Allowable Classes and Deco

allowables.py

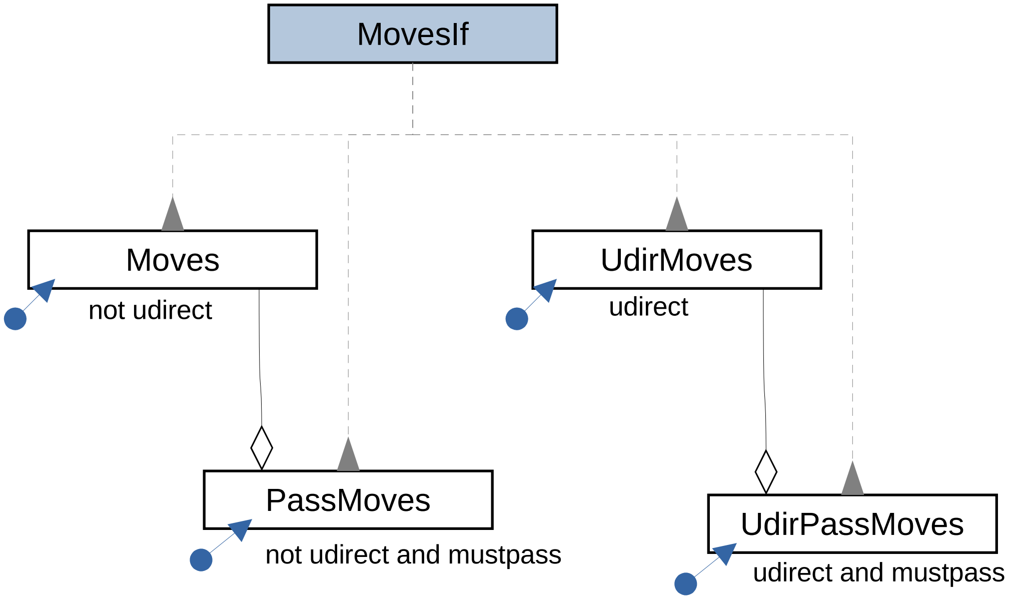


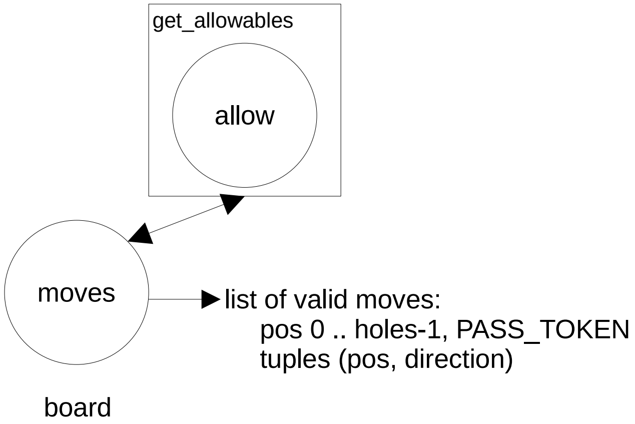
Possible decorator chain start

Δ game state changed

Get Moves Classes and Deco

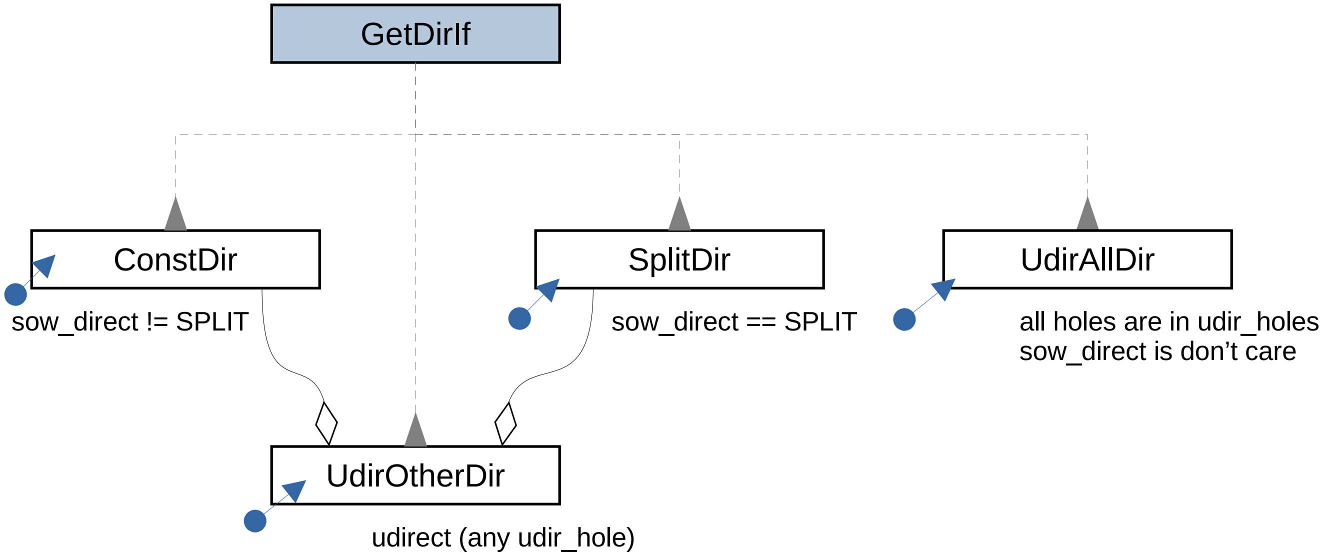
get_moves.py





Get Direction Classes and Deco

get_direction.py



GetDirIf

ConstDir

SplitDir

UdirAllDir

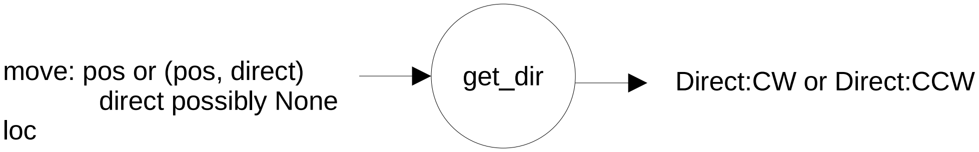
UdirOtherDir

sow_direct != SPLIT

sow_direct == SPLIT

all holes are in udir_holes
sow_direct is don’t care

udirect (any udir_hole)



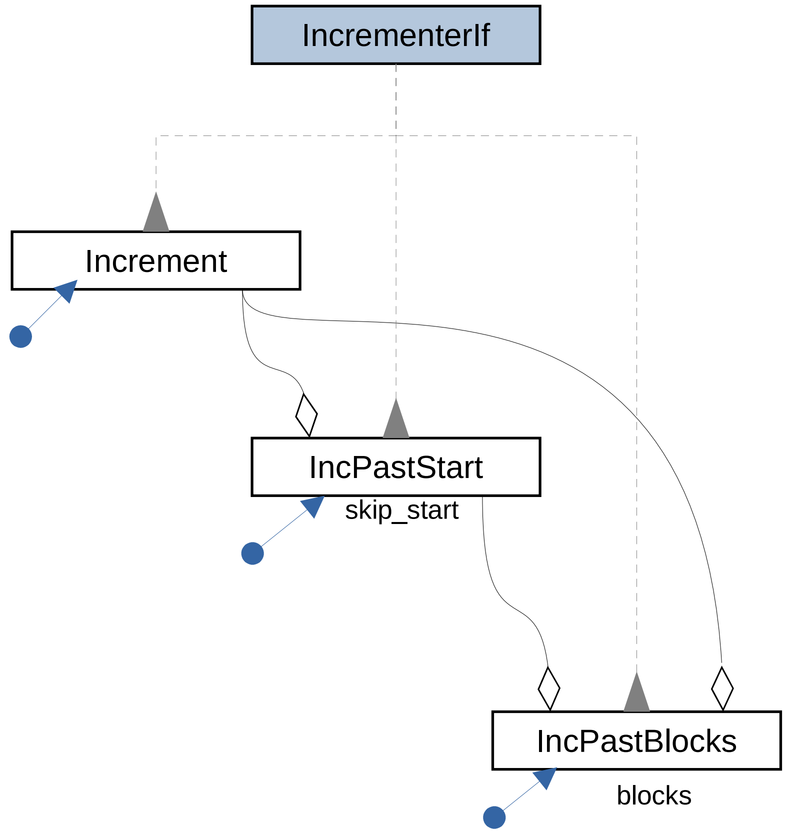
move: pos or (pos, direct)
direct possibly None
loc

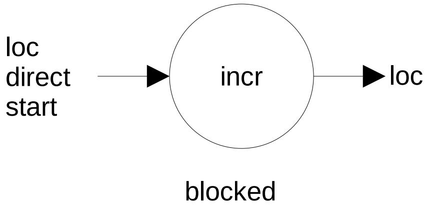
get_dir

Direct:CW or Direct:CCW

Incrementer Classes and Deco

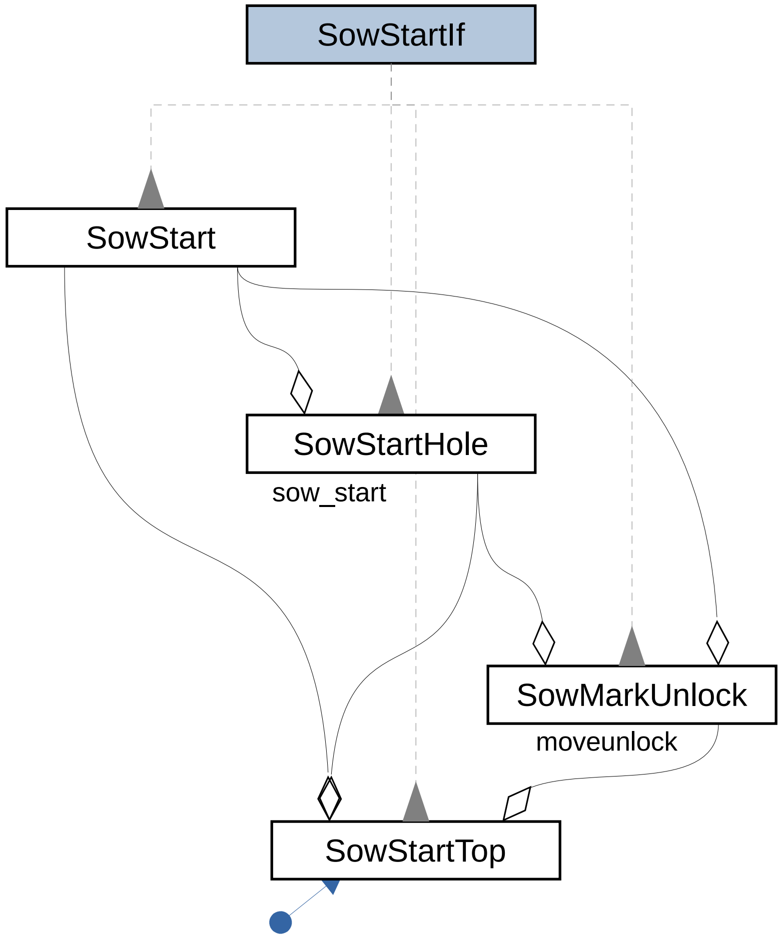
incrementer.py

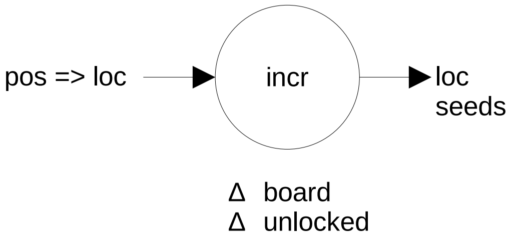




Sow Starter Classes and Deco

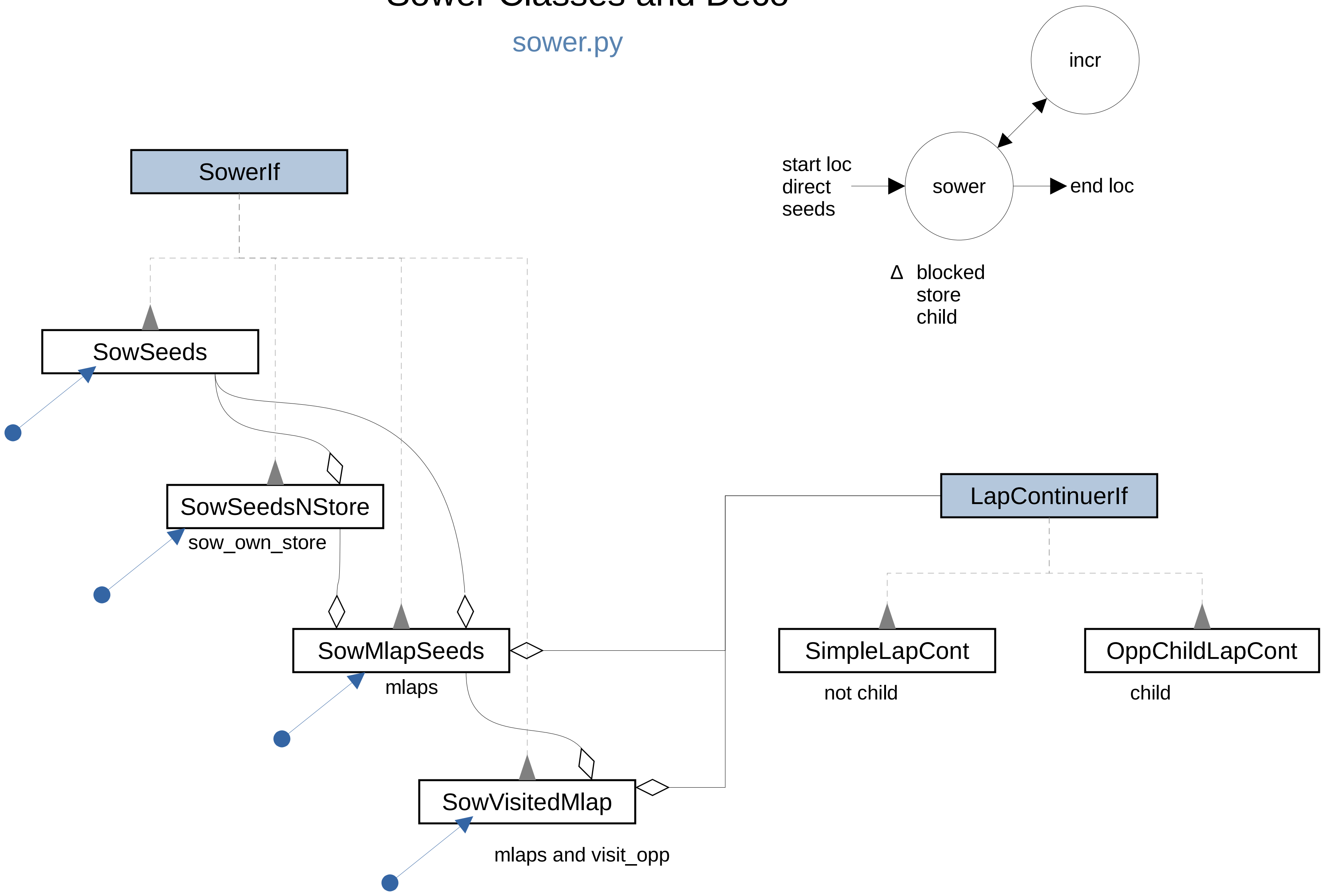
sow_starter.py





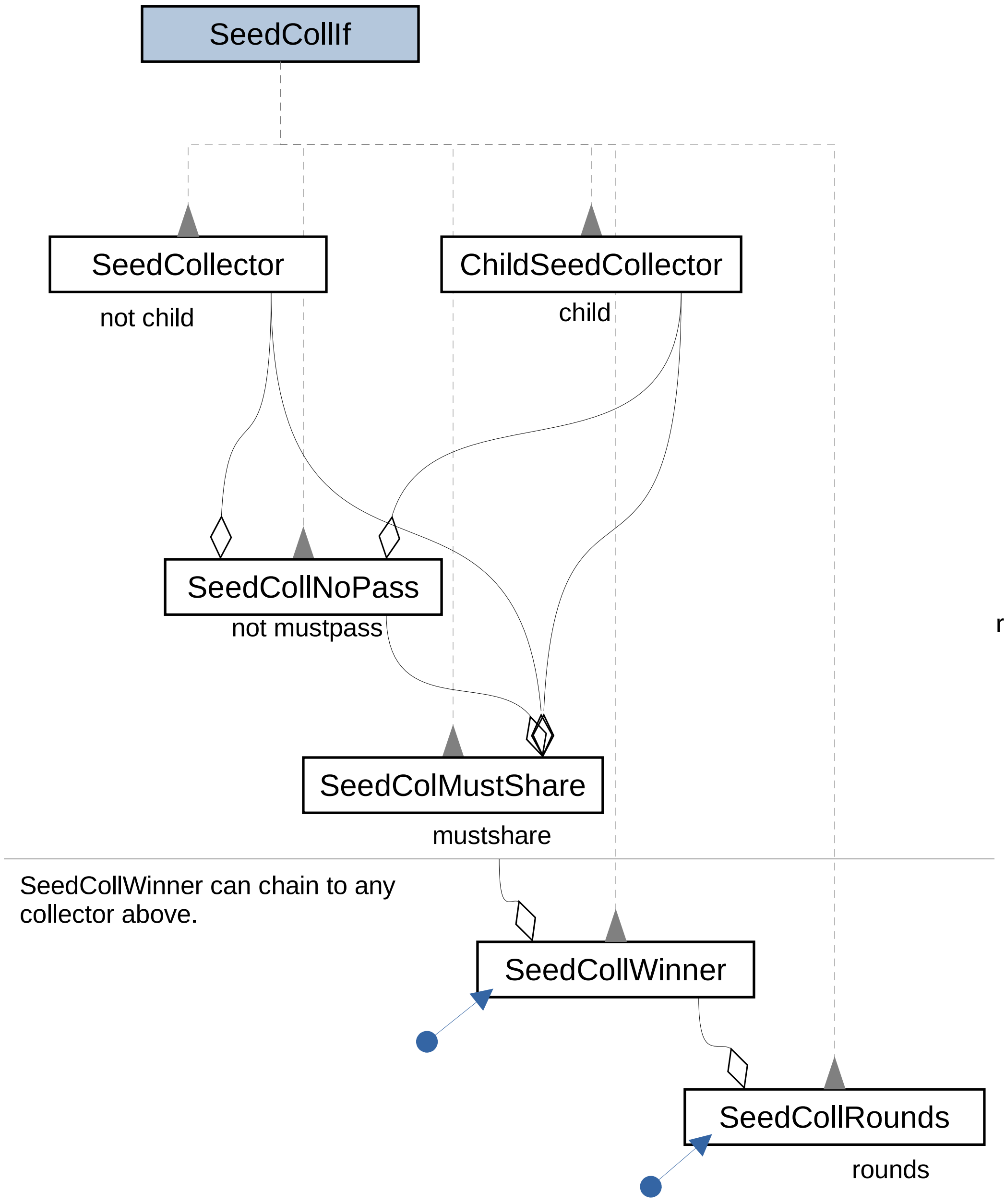
Sower Classes and Deco

sower.py



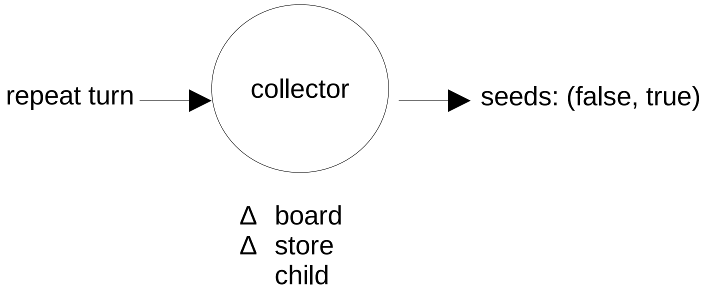
Seed Collector Classes and Deco

seed_collector.py



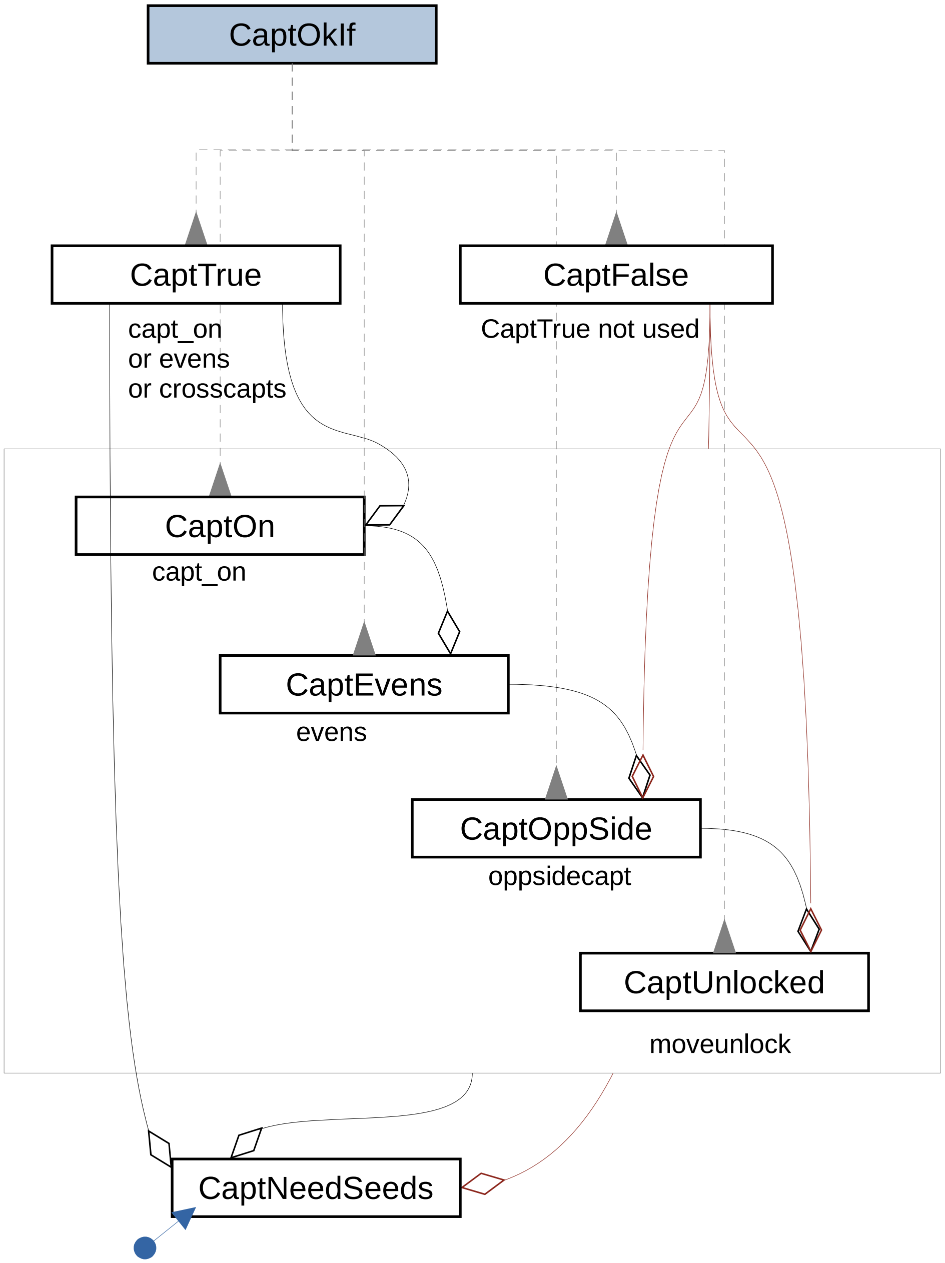
SeedCollWinner can chain to any collector above.

The seed_collector deco chain is used as a first step in determining if the game or round is over and/or there is a win/tie. Each deco can decide that the game is over.



Capt Ok Classes and Deco

capt_ok.py



CaptOkIf

CaptTrue

capt_on
or evens
or crosscapts

CaptFalse

CaptTrue not used

CaptOn

capt_on

CaptEvens

evens

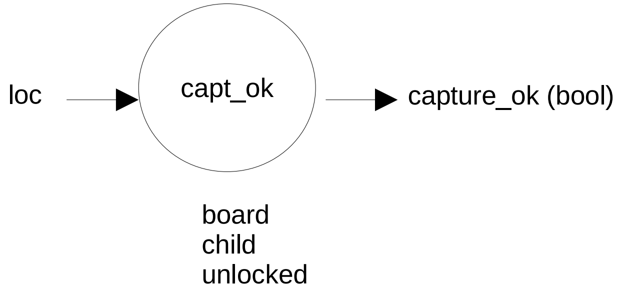
CaptOppSide

oppsidecapt

CaptUnlocked

moveunlock

CaptNeedSeeds



loc

capt_ok

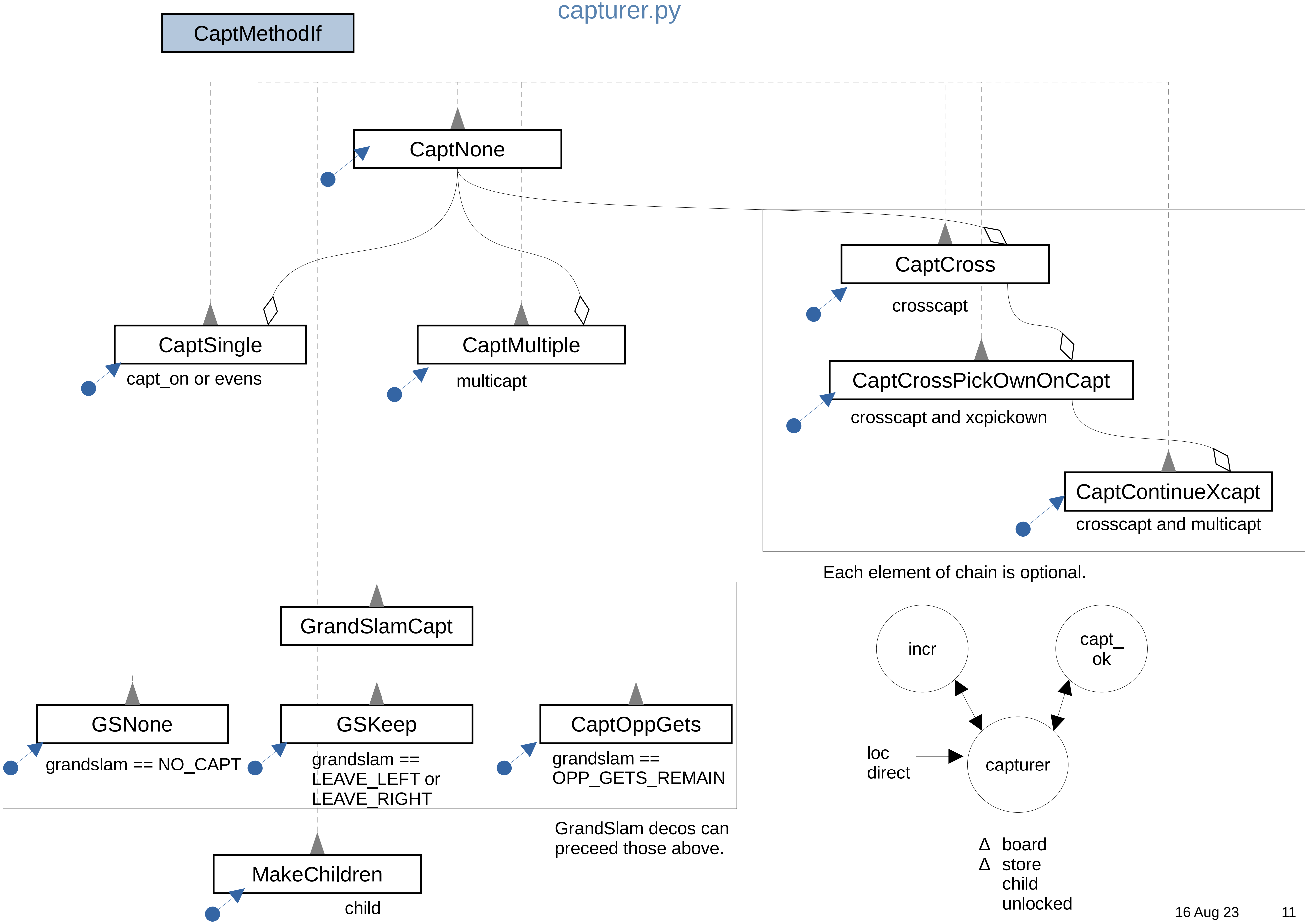
capture_ok (bool)

board
child
unlocked

Each element in chain is optional based on specified conditions.
The link goes to the next deco that applies.
The deco chain that ends in CaptTrue are effectively anded together.
The CaptFalse doesn’t really make sense.

Capturer Classes and Deco

capturer.py



CaptMethodIf

CaptNone

CaptSingle

capt_on or evens

CaptMultiple

multicapt

CaptCross

crosscapt

CaptCrossPickOwnOnCapt

crosscapt and xcpickown

CaptContinueXcapt

crosscapt and multicapt

Each element of chain is optional.

GrandSlamCapt

GSNone

grandslam == NO_CAPT

GSKeep

grandslam ==
LEAVE_LEFT or
LEAVE_RIGHT

CaptOppGets

grandslam ==
OPP_GETS_REMAIN

GrandSlam decos can
preceed those above.

MakeChildren

child

incr

capt_
ok

loc
direct

capturer

Δ board
Δ store
child
unlocked