smartPlayer

--All the other helper functions are included in three functions below

--Some functions have two Hands but can be regarded as one hand

checkScoreDom

A function that may return a domino that can make me win

When there is a domino that makes me win

checkScoreDom [(5,3),(2,2),(6,0),(6,2),(5,4),(2,0),(1,1),(3,1)] P1 (Board (4,4) (4,2) [((4,4),P2,2),((4,2),P1,1)]) (57,43)

Just ((2,2),R)

When there is no dominoes that make me win

checkScoreDom [(5,3),(2,3),(6,0),(6,2),(5,4),(2,0),(1,1),(3,1)] P1 (Board (4,4) (4,2) [((4,4),P2,2),((4,2),P1,1)]) (57,43)

Nothing

When my score is less than 53 (always Nothing)

checkScoreDom [(5,3),(2,2),(6,0),(6,2),(5,4),(2,0),(1,1),(3,1)] P1 (Board (4,4) (4,2) [((4,4),P2,2),((4,2),P1,1)]) (50,43)

Nothing

Safe59

A function that may return a domino that makes me score 59 when it is guaranteed that I have all the dominoes that make the opponent win after my move

When there is a domino that makes me become 59 when I have all the dominoes that makes the opponent win after my move

safe59 [(0,3),(2,5),(6,4),(5,4),(2,0),(1,1),(3,2)] [(0,3),(2,5),(6,0),(6,4),(5,4),(2,0),(1,1),(3,2)] P1 (Board (6,6) (4,2) [((4,4),P2,2),((4,2),P1,1)]) (57,58)

Just ((6,4),L)

When there is a domino that makes me become 59 but I don’t have all the dominoes that make the opponent win after my move

safe59 [(5,3),(2,2),(6,2),(5,4),(2,0),(1,1),(3,2)] [(5,3),(2,2),(6,0),(6,2),(5,4),(2,0),(1,1),(3,2)] P1 (Board (6,6) (4,2) [((4,4),P2,2),((4,2),P1,1)]) (57,58)

Nothing

When I don’t have any domino to make me become 59

safe59 [(0,3),(2,5),(5,4),(2,0),(1,1),(3,2)] [(0,3),(2,5),(6,0),(5,4),(2,0),(1,1),(3,2)] P1 (Board (6,6) (4,2) [((4,4),P2,2),((4,2),P1,1)]) (57,58)

Nothing

noBust

A function that may return a domino that does not exceed 61 and I have all the dominoes that can make the opponent win after my move. (will be combined with the condition of my score>53)

When there is a domino that does not exceed 61 and have all the dominoes that can make the opponent win after my move

noBust [(6,0),(2,6),(0,4),(1,1)] [(6,0),(2,6),(0,4),(1,1)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (59,59)

Just ((6,0),L)

After my play (6,0), the opponent need (0,4) and (2,6) but they are already on my hand, so I can play this domino.

When there is a domino that does not exceed 61 but not have all dominoes that can make the opponent win after my move

noBust [(6,0),(5,4),(1,1)] [(6,0),(5,4),(1,1)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (59,59)

Nothing

When there is no domino that does not exceed 61

noBust [(2,3),(5,4),(1,1)] [(6,0),(5,4),(1,1)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (59,59)

Nothing

Blocking

A function that may return a certain domino that will block the opponent to win after my move (will be combines with the condition of opponent score>=53)

When there is a certain domino that will prevent the opponent winning after my move

blocking [(6,3),(2,0),(6,5)] [(6,3),(2,0),(6,5)] (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) P1 (50,60)

Just ((2,0),R)

When there is no certain domino that will prevent the opponent winning after my move

blocking [(4,3),(2,0),(6,5)] [(4,3),(2,0),(6,5)] (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) P1 (50,60)

Nothing

prevent59

A function that may return a domino that prevents the opponent from getting score 59 after my move

All dominoes that make the opponent to get 59 are on the board and the hand after my move

prevent59 [(6,6),(2,0),(0,4),(0,6)] [(6,6),(2,0),(0,4),(0,6)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (50,57)

Just ((6,6),L)

Not all dominoes that make the opponent to get 59 are on the board and the hand after my move

prevent59 [(2,0),(0,4),(0,6)] [(2,0),(0,4),(0,6)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (50,57)

Nothing

get59

A function to make me score 59

When there is a domino to become 59

get59 [(2,0),(0,4),(0,6)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (57,57)

Just ((2,0),R)

When there is no domino that makes me become 59

get59 [(0,4),(0,6)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (57,57)

Nothing

When my score is less than 53 (always Nothing)

get59 [(0,4),(0,6)] P1 (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)]) (50,57)

Nothing

smt

Safe domino means that after my move all the dominoes that can make higher score than the score I made are on my hand or the board. i.e. there is no possibility for the opponent to get higher score than me

If there is a safe domino then return the safe domino

If not

If there is 70% domino then return the domino

If not

Just return the highest score dom

When all the dominoes are on the board and hand that makes the opponent can get higher score than the score after my move

smt [(2,0),(0,3),(6,6),(0,6)] (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)])

safe dom

((2,0),R)

If there is a dom that makes the opponent score less than me, this domino is considered to be safe.

When there is no safe domino than get the highest score domino

smt

smt [(2,0),(0,4),(0,6)] (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)])

high score dom

((2,0),R)

When (6,0) and (6,3) are at the hand with Init Board

smt [(2,0),(0,3),(6,6),(0,6)] InitBoard

((6,6),L)

When there is (5,4) on my hand with InitBoard

smt [(2,0),(5,4),(6,6),(0,6)] InitBoard

((5,4),L)

When there is no special case (the, get the highest score dom)

smt [(2,0),(3,6),(5,2)] InitBoard

((3,6),L)

Morechanece

Play my domino and then get all the dominoes that can get higher score than I made. Check how many these dominoes are on board++hand and then divide by total number of potential higher scoring dominoes. If the percentage higher, then 70 then return the domino.

When there is a domino that are 70 percent safe (7 dominoes out of 9)

morechance [(2,2),(5,4),(5,3),(5,1),(2,1),(5,5),(5,6)] [(2,2),(5,4),(5,3),(5,1),(2,1),(5,5),(5,6)] (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)])

Just ((5,6),L)

When there is a domino but less than 70 percentage safety

morechance [(2,2),(5,4),(2,1),(5,5),(5,6)] [(2,2),(5,4),(2,1),(5,5),(5,6)] (Board (6,4) (4,2) [((6,4),P2,2),((4,2),P1,1)])

Nothing

All the functions are tested above

LOGIC

A) checkScoreDom -> check I can become 61 to win the game

if not

B) safe59 -> check there is a domino to become 59 that does not lead to the opponent’s victory.

if not

C) (oppscore>=53)|| (myscore>=53)) && noBust -> find the domino that does not exceed 61 and check that is safe domino. If both players score are less than 53, then it is better to get the domino that can make higher score than the opponent. (opponent can’t win).

If not

D) (oppscore>=53)|| (myscore>=53))&& blocking -> pretty much same as noBust, but when my score is bigger than 53 then there can be possibility of the safe domino that exceed 61 that makes the opponent not to win.

If not

E) (oppscore>=51) && prevent59 -> prevent the opponent to get 59 (more possibility to win the game).

If not

F) Get59 -> Just try to get score 59

If not

G) Smt-> so mostly if my score and the opponent score is less than 53 (before end game), (When two people are less than 53 then tactic is always getting higher score than the other

)

Basic tactic is to, considering the board and the hand, find out the safe dominoes. If there is no safe one, then just return 70% safe dominoes. If there is still nothing, then just play the highest domino score

When InitBoard, if the player has (6,6) and (0,3) then play (6,6) and if there is (4,5) then play this domino. This is because the opponent cannot get higher score than I did. 🡪This does not give better result so commented out.

Seed42 and 1000Games play agians hsdPlayer

So if I Play hsdPlayer – 498 win

If I play smartPlayer only with tactic G – 525 win

If I add tactic A – 615 win

If add tactic B – 628 win

If add tactic C -654 win

If add tactic D =655 win

IF add tactic E – 662 win

If add tactic F – 665 win

If add InitBoard tactic = 658 win

So every time new function added, the winning time also goes up

Therefore it can be said smartPlayer capture much knowledge for the domino game