Assignment 3 Domino Players Design

This design contains the main tactics I have designed and implemented, note in some cases only includes the main functions that a tactic employs and some functions that a tactic uses may use some more trivial functions to reach their state. Also included is a design for calculating the opponents hand, and at the end a design and description of how I will use tactics in conjunction with one another to define a type of Player.

Data Types

|  |  |
| --- | --- |
| **Data type** | **Description** |
| type Dom = (Int,Int) | Data type for a domino represented as a tuple |
| data DomBoard = InitBoard|Board Dom Dom History | Data type representing the state ends of the board and the history of play |
| type Hand = [Dom] | Data type representing a list of dominoes |
| data Player = P1|P2 | Data type to decipher a player turn |
| type MoveNum = Int | Type representing a number associated with a move |
| type History = [(Dom,Player,MoveNum)] | List of tuples representing who played each dom and win |
| data End = L|R | Data type used to specify an end of the board |
| type Move = (End,Dom) | Data type representing a move which returns a domino to play and the end to play it at |
| type DomsPlayer = Board -> Hand -> Move | A datatype which takes a function, allows for different domino players to be created. A domino player will take a hand and a board and return a move to make. |
| type GameState =(Hand,Hand,Player, DomBoard, Scores) | state in a game - p1's hand, p2's hand, player to drop, current board, scores |
| type DomsPlayer = Hand->DomBoard->Player->Scores->(Dom,End) | given a Hand, the Board, which Player this is and the current Scores returns a Dom and an End |

**firstDrop tactic**

Plays optimal scoring (5,4) domino if player has first go, when playing this dom the score gained when placing is greater than any

possible response the opponent can play

Move

Hand,Domboard

firstDrop

Play (5,4) if player has it and board is empty

**Winning Move Tactic**

Gets the move of the winning domino if the player has a domino in their hand that can win them the game



MoveThatWillScore

Gets a move from a hand that will score a specific score

Move

Int,DomsPlayer

Int

Int

ScoretoWin

Gets score required to win

Int

Player,Scores

GetPlayerScore

Gets the score of theplayer

moveThatWillWin

Get move of winning dom

Move

DomsPlayer

**Scoring 59 tactic**

Reaching a score of 59 is a good tactic since there are many ways of scoring 2 points, so if 59 can be reached a player is more likely to win on their next turn.



MoveThatWillScore

Gets a move from a hand that will score a specific score

Move

Int,DomsPlayer

Int

Int

Scoreto59

Gets score required to score 59

Int

Player,Scores

GetPlayerScore

Gets the score of theplayer

Move

DomsPlayer

moveThatWillScore59

Get move of dom that will score 59

**Move that stops opponent from winning**

If the player can win based on what is believed to be in their hand when I play a dom then don’t play it, find a dom that when played will stop opponent from a win based on what is in their hand. Gets a list of doms from a hand that will stop opponent from winning and then plays the highest of those doms.

**See next slide for**

**getting OPP hand**

Hand

Hand,Domboard

opponentsPossHand

Gets a list of doms that opponent could possibly have in their hand

getWinDoms

Gets a list of doms that opponent could win with if I place that piece

Hand

Hand,Domboard,Player,Scores

Boolean

Dom,Hand,Domboard,Player,Scores

canOpponentWin

Returns a bool as to whether the opponent can win if a place a specific piece

Hand

checkOpponentWin

Get move of dom that will prevent opponent winning

Hand,Domboard,Player,Scores

**Get Opponents Possible Hand**

Generates a hand that the opponent could have based on what is not in the players hand, what has been played and what the opponent was knocking on in the past.

opponentsPossHand

Gets a list of doms that opponent could possibly have in their hand

Hand

Hand,Domboard

Hand, Int

domsWithValue

Gets a list of the doms from domset that have a certain spot value

History,Hand

**See next slide for calculating when player was knocking in history**

Hand

(Int,Int)

History

endvalsKnockingOn

Gets a tuple pair of the values the player was knocking on if they were knocking

Hand

domsInHistory

Gets a list of the dominoes that have been placed

**Get Opponents Possible Hand (continued)**

endvalsKnockingOn

Gets a tuple pair of the values the player was knocking on if they were knocking in history

(Int,Int)

Hand

History

constructBoard

Returns the Board of the reconstructed history before player was knocking

History,Hand

History,Int

History

Int

reconstructHistory

Reconstructs the history of before the player was knocking

History

moveKnockingOn

Gets the move number opponent was knocking on

History

History

History

History

sortHistory

Sorts history by move number

knockingOn

Returns the history of when the player was knocking

**Play highest scoring domino unless opponent can score more afterwards**

Uses a construction of the opponent’s possible hand to simulate the highest scoring dom they could play for each of domino players highest scoring doms can play.

hsdPlayer

gets highest scoring move from hand

DomsPlayer

Move

Move

DomsPlayer

Hand

Hand,Int

opponentsPossHand

Gets a list of doms that opponent could possibly have in their hand

opponentDoesNotScoreMore

Returns the move of dom player can play that will not result in opponent scoring more on their next turn

Function will Simulate a move based on players highest scoring dom of current domboard and then simulate move based on opponents highest scoring dom of the new domboard. If opponents highest scoring scored less than my highest scoring then will return move of my highest scoring.

scoreboard

5s & threes score for a board

Int

DomBoard

Dom,End,Player,DomBoard

DomBoard

updateBoard

Return new domboard after a play

**Play highest scoring unless can’t knock off**

Function will play my highest scoring domino so long as it is not a dangerous dom and player can knock it off with another dom in players hand. (dangerous dom is considered to be any double dom)

Dom,Int

Bool

Bool

domContainval

returns true if a dom contains a spot value

Domswithval

Gets a list of doms from a hand that contain a certain spot value

Hand

Hand,Int

Dom

isDouble

returns true if domino is a double

Move

DomsPlayer

hsdPlayer

gets highest scoring move from hand

highestKnockOff

Returns the move of my highest scoring domino if its double and it can be knocked off

Move

DomsPlayer

**Play Majority Dom**

If a player has a majority of one spot value then play one of those doms, if the double of that spot value is in this majority list then play that first, otherwise play the highest scoring that will go of that majority spot value.

isMajority

returns true if a dom contains a spot value

Bool

Hand,DomBoard,Int

Hand,Int

Hand

Hand

Dom,Int

Bool

domContainval

returns true if a dom contains a spot value

Domswithval

Gets a list of doms from a hand that contain a certain spot value

Hand,Domboard

getmajoritylistfromhand

Returns list of doms that are a majority of one spot value from a hand

Hand,Domboard

Move

playMajority

Returns move of highest scoring dom that is in a majority of doms I may contain

**Play Majority Dom (continued)**

Bool

isMajority

returns true if a dom contains a spot value

Hand,DomBoard,Int

History,Hand

History

DomBoard

getHistory

returns history from a domboard if there is one

domsInHistoryOfValue

gets a list of all the doms in the history by recursing through it

Int

DomBoard,Int

domsInHistoryOfValue

gets the number of doms with a spot value played on the board

Hand

**Example player playing a combination of tactics**

Below Is a design of what a player could look like when using a combination of different tactics, simply put if a tactics conditions are not met then they will return an empty sentinel value. If a player has gone through all their tactics and none of their conditions are met then they will all return this sentinel value. In this case the player will just result to playing its highest scoring domino.

DomsPlayer

DomsPlayer

Hand,Domboard,Player,Scores

DomsPlayer

Move

Move

Move

Move

1. moveThatWillWin

Get move of winning dom

2.moveThatWillScore59

Get move of dom that will score 59

3.checkOpponentWin

Get move of dom that will prevent opponent winning

4.hsdPlayer

gets highest scoring move from hand

cleverPlayer

Goes through its tactics in order, if none can be used uses HSDPlayer

DomsPlayer

Move