Dominoes Assignment 3 Briefing

The aim is to programme smart Domino Players

The Haskell mechanism for playing a game is provided for you

All your players have to do is decide which Dom to play at which end

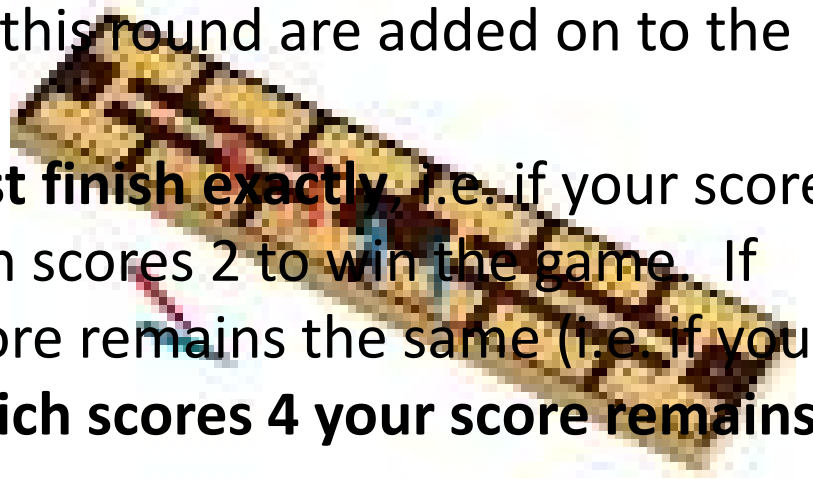
You can experiment by having your players compete amongst themselves

Assignment 3 is 25% of the credit for COM2001

A Dominoes Match

A 2-player 5s-and-3s dominoes match is organised as follows:

- A match consists of N games, typically 3 (in human play). The winner is the player who has won most games.
- Each game involves a sequence of rounds (as defined in assignment 2). The two players take it in turns to drop first in each round.
- A game terminates when one player or the other wins by achieving an accumulated score of **exactly 61**, marked out on a 'pegging board'
- If at the end of a round neither player has reached 61, a new round starts and the scores obtained within this round are added on to the scores after the last round.
- **As with all good pub games, you must finish exactly**, i.e. if your score is 59 you need to play a domino which scores 2 to win the game. If you score more ('going bust') your score remains the same (i.e. if your score is 59 and you play a domino **which scores 4 your score remains at 59).**



Hints about skilled play

- Some players normally play the highest scoring dom unless it risks the opponent scoring more (e.g. if the ends are 6 and 0 and your hsd is (6,6), play HSD unless the opponent could play (0,3) or (0,6).
- Beware of playing a dangerous Dom unless you can knock it off, e.g. playing (6,6) if there are no more 6s in your hand.
- The History tells you what doms remain and therefore what to guard against.
- You can use the History to work out what doms your opponent may have (e.g. if the ends 5 & 5 & the opponent didn't play (5,5) s/he doesn't have it.
- Knowledge about what your opponent is knocking on is particularly useful.
- If you have the majority of 1 spot value (e.g. 4 6s) it's a good idea to play that value, especially if you have the double.
- If you have a weak hand, try to 'stitch' the game, i.e. make both players knock.
- If you are getting close in the end game, you should obviously see if you have a winner. If not, try to get to 59, because there are more ways of scoring 2 than scoring 1,3 or 4.
- If the opponent is getting close in the end game, look for a dom which will prevent her/him winning on the next play, or reduce the chances of this happening.
- If you have 'first drop' onto an empty board, a popular choice is (5,4), because it scores 3 but the maximum you can score in reply is 2.

Designing the Doms Player

You are free to programme your Doms players anyway you like provided you start with, and submit, your design. However..

- Except in the End Game, search techniques won't get you far (what's the goal state?)
- Instead you want to base your players on the hints (and others you may think of)
- Each hint you use should be coded separately into a Tactic
- A Tactic will be associated with a game situation
- You need to be able to easily change your Tactics
- There is a natural order to some Tactics
- A player can be defined by which Tactics it uses, in which order
- You should design a framework along these lines before you code any players.

Christmas Competition

- In week 12 I'll invite you to submit your best player to me
- I'll run a Round Robin competition (i.e. every player plays every other player
- Prize for the winner
- Not compulsory & not worth marks!