COMP1140 Structured Programming(Advanced)

Wed Aug 22,2018

Design of the game(Skeleton)

Tutorial Group:Thursday 2-4pm

University ID:u6683369,u6684916

ANU College of Engineering & Computer Science

Introduction

In this assignment we will write code for the blooms game including the game, UI and AI, here's the skeleton of our game(initial design, might be changed later while we are doing the assignment).

Game(BloomsGame class)

For this part, we need to finish the task2-11 which are isPiecePlacementWellFormed, getNeighbours, getBloom, isFenced, isPlacementValid, getScore,generateMove functions.

For task3, we use 3 boolean values which are b1,b2,b3 to determine the 3 conditions respectively(consists of 2 or more characters or not, whether the first character is a letter representing colour, whether the remaining characters represent a natural number). In this particular task, we use substring,split and matches functions, the substring function is to cut off the string and get the particular part that we would like to get by the index. As for the split function, it's kind of similar to the substring function, but this time, instead of split to string by the index, we split it by a particular feature, like whether the string contains letters "a" to "d". The matches function is to check the particular element is in the our desired form or not. And you are going to see them a lot in this assignment.

For task4, we get some idea from the website of the game design(which is given by the assignment webpage). The main idea is to build an two-dimensional array to give coordinates to get its neighbours but except one situation which is the board can't have any neighbours(in size of 1 or less). Although the return type is in the form of the array of int, but because we don't know the size of the array first(neighbours' number change as the position on the board changes) so we put the results in a list. That concept is also going to be used frequently in this assignment.

For task5, we need to build an UI for the game to visualize the placement of the board. So I am going to make a field called stone to signify the stones on the board, and make pictures of the stones colored as a,b,c,d. First I am going to record all positions on the board and mark the index on them from 0-36, then according to the placement of the string, I put all the stones to the position as the placement asked. That's the intial version of our UI.

For task6, we need to getblooms of the current board, first we separate the board into numbers and letters, then using numbers we can get the colors of the numbers because numbers and letters share the same index numbers. After that, we look back to the index number before that space to find the spaces which share the same colour and put them in a new list(onecolorlist), for the result list, we just need to transform this colorlist to a list of the numbers according to their index numbers and concatenate the colour and the number according to it together.

For task7, we need to determine whether a given bloom is fenced, first we separate the board into numbers and letters. Then we use n-2 array to build a map of the board, if the space is placed outside out map, then we use letter "e" to mark it. Also if the space is in the bloom that we would like to get then we also mark it. Next, by using getNeighbours function that we have written previously, we can put the place around the bloom into a new list. Then we judge whether the bloom is fenced or not, if the surrounding spaces have our color or empty then it suggests that we are not fenced so we return false, otherwise is true.

For task8, first we need to check whether the piece placement is well formed or not, then we separate it to 3 situations:o placement, 1 placement or 2 placements. If it's 0 placement, then it's automatically valid so we return true straightaway. Then we need to check for 1 placement by finding if that's fenced or not and similar method apply to 2 placements.

For task9, first we need to count how many player1's stone are on the board and how many player2's stones are on the board. Then we need to get empty spaces by get all spaces on the board and get rid of the places which are already on the board. Then we need to use getNeighbourNew

folder function to get all neighbours' space on the board hence find its colour. If all surrounding colours belong to player1, then player1's score plus one and vice versa.
New folder
For task11, I think there are some issues with it, we need to output all possible moves instead of one single move. We can see whose turn is by the movesequence string and generate all moves, by applying isValidMove to the all moves, we can hence get all possible legal moves for the current board.

Remark: You can find all codes for task3-9 in our assignment 2 so far.

Board

In this section, we are going to get all parts of the game working on our laptop by using all functions.

The method that we are plan to build:New folder

1. public static String updateBoardString(String placement)
   Every time when player make a movement add to the board also sort the board string by number order eg.a2b3c1 → c1a2b3 return new board string if(isPlacementValid){updateBoard(..)}

2. public static String removeFenced(String board) check every placement and remove fenced placement return new board string need to check every placement that is fenced other than the given one and then update

3. public static int isTerritory(int size, String board, int space) return 1 for player 1, 2 for player 2 and -1 for not a territory.

4. public static boolean isFinished(int size, String board) only empty space is one's territory return true this method start checking after 15 placements have made

   second thought: when player have made "." placement more than three times may have a function record placement and checking from backwards.

AI

In this part, we plan to build AI in 3 different difficulty levels, the beginner level is implemented by choosing random move from all possible move that the player can take in this turn. The more advanced level we are planning to build is to use a strategy called maxium the score, in this AI, the strategy that we are going to take is just to make the difference between AI and player as big as possible. Then the expert level we are going to use the alpha-beta pruning to verse the player.

The functions that we are going to use in this part are randommoves which is to pick a random move from all possible moves, the second function that we are going to use in this part commonly is makemove which similar to the assignment 3 of COMP1130 that we did last semester, which is to make the computer and player make moves. Then we might build two classes for maxiumscore and alpha-beta pruning respectively.

References

Hexagonal Grids, https://www.redblobgames.com/grids/hexagons/

How to sort an arraylist, https://stackoverflow.com/questions/16252269/how-to-sort-an-arraylist

Concatenate integer arraylist to a string, https://stackoverflow.com/questions/599161/best-way-to-convert-an-arraylist-to-a-string