GRP_30: All Clear Solver

Greg Wang (19GBW1)

Steven Zhang (19SZ99)

Johnathan Wilson (19JMW19)

Brian Grigore (19BG1)

Course Modelling Project

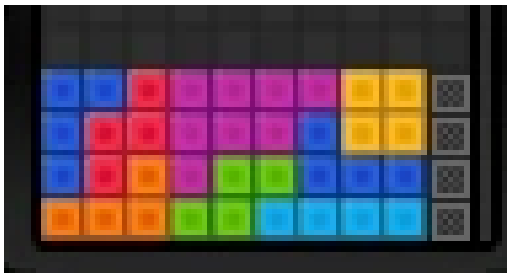
CISC/CMPE 204

Logic for Computing Science

December 14, 2021

Abstract

Tetris is likely the most recognizable video game in history, known for its very simplistic, easy to understand premise, though digging deeper there is a lot more to this game than first meets the eye. This application will solve in tetris what is known as an all clear, which is when, after the first move, the pieces on the board are all clear leaving an empty board behind. For the sake of terminology, when we discuss pieces later in this assignment we will use the word "tetromino" as that is the official term. Our model will detail how the all clear will be created by documenting the origin position and rotation of each tetromino provided by the user. Our model after beign generated will then be displayed to the user to show the board state achieved by the all clear. The rotational states of all tetrominos is referred to as the piece's SRS state (which stands for Super Rotation System state) which will be used when referring to a piece's configuration.



Seen above: An all clear being set up with 1 O piece, 2 J pieces, 2 I pieces, 1 S piece, 1 L piece, 2 T pieces and 1 Z piece.

Our model will not be taking into account the order of the pieces and actually having them fall into place, instead we will be simply placing them into the correct position to achieve an all clear. Our model will be taking into account every single permutation of all 10 pieces given. The number of permutations will depend on the number of possible SRS states of the tetromino and the anchor positions possible in that SRS state.

Propositions

f_{ij}: This is true when tile (i,j) (Counting the bottom left as 1,1) is filled by a unit of a tetromino
e_{ij}: This is true when tile (i, j) is empty and not filled by a unit of a tetromino

c_i: This is true when row i is filled with pieces and is cleared

O_{ijkl}: This is true when an O piece is anchored at (i, j) with SRS state k and with id l (since more than one of the same piece can be used at once and id is used to separate individual pieces)

I_{ijkl}: This is true when an I piece is anchored at (i, j) with SRS state k and with id l (since more than one of the same piece can be used at once and id is used to separate individual pieces)

L_{ijkl}: This is true when an L piece is anchored at (i, j) with SRS state k and with id l

J_{ijkl}: This is true when an J piece is anchored at (i, j) with SRS state k and with id l

S_{ijkl}: This is true when an S piece is anchored at (i, j) with SRS state k and with id l

Z_{ijkl}: This is true when an Z piece is anchored at (i, j) with SRS state k and with id l

T_{ijkl}: This is true when an T piece is anchored at (i, j) with SRS state k and with id l

P_i: Is true when a piece is type i

N_i: The number of pieces active

Constraints

We want to model an all clear with the pieces we're given with a board with dimensions $\{(i, j)|1 \leq i \leq 10, 1 \leq j \leq 4\}$ since we only want to place and fill in the 10x4 bottom of the grid since we are using all 10 pieces to achieve a perfect clear and each piece take up 4 spaces, this means:

$\forall i \forall j(\neg(f_{ij} \wedge e_{ij}) \wedge \neg(\neg f_{ij} \wedge \neg e_{ij}))$: No tile can be both empty and filled at the same time and no tile can be neither (XOR).

$\forall i \forall j \forall k \forall l(O_{ijkl} \rightarrow P_O)$: Occurs for all tetromino types, where if one of that tetromino type then the corresponding P proposition for that type is true.

$\forall j(f_{1j} \wedge f_{2j} \wedge f_{3j} ... f_{10j} \rightarrow c_j)$: If a row has all it's tiles filled, the row is cleared.

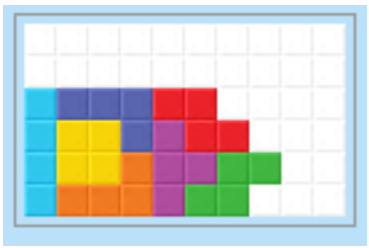
$O_{1200} \rightarrow ((e_{12} \wedge e_{22} \wedge e_{11} \wedge e_{21}) \rightarrow (f_{12} \wedge f_{22} \wedge f_{11} \wedge f_{21}))$: This constraint is the necessary constraint to not allow overlap of pieces as every piece being placed will need the spaces to be all empty first before being able to fill out the spaces dictated by the tetromino's anchor position and rotation.

$N_{10} \rightarrow \forall i \forall j(f_{ij})$: Once all of the pieces have been placed, then all 40 tiles will have been filled up.

Model Exploration

When exploring our model we used a text file of known full clear solutions that contained known solutions and the pieces needed to create that solution so we planned to use that to check against our All Clear Solver. The text file contained

some obvious solutions, such as 10 I pieces, 10 O pieces and some deviations of a standard all clear setup found below which can create an all clear with just a few more pieces.



During the intermediate stages of our testing we tried to fix many of the bugs with overlapping pieces and only allowing pieces to be placed if all squares for that piece were empty, though unfortunately we ran into errors and while doing our best to debug, didn't end up getting bauhaus to work the way we wanted to.

This was mainly due to some limitations we ran into while coding we weren't sure how to get around so ultimately we couldn't fix every bug in time.

Jape Proof Ideas

$x \to (\neg z1 \land \neg z2 \land \neg z3), y1 \to z1, y2 \to z2, y3 \to z3, x \vdash \neg(y1 \land y2 \land y3)$:
Proving that if all a pieces tiles fall into a line being cleared, the piece does not exist anymore

$(P(x) \to (C(x) \land \exists y.(R(y) \to \neg R(y)))), P(x), \forall x.R(x) \vdash \exists x.\neg R(x)$:
Once a line is cleared, there will always be a row of tiles that gets shifted down because of the clear

$N(x) \to \forall i.\forall j.F(i,j), N(x) \to \forall i.\forall j.C(i,j), N(x) \vdash \forall i.\forall j.C(i,j) \to \forall i.\forall j.F(i,j)$:
An all clear at once is the same effect as clearing all 4 rows individually

$N(x) \to \forall i.\forall j.F(i,j), N(x), \forall i.\forall j.F(i,j) \to \forall j.C(j), \forall j.C(j) \to \forall i.\forall j.E(i,j) \vdash \forall i.\forall j.E(i,j)$:
An all clear also leaves every tile empty at the end after filling every tile

First-Order Extension

Converting the Propositions: These will mirror the propositions stated above. We have objects in the first order setting that correspond to coordinates, rotation, and identification code.
$Coord(i,j)$: objects i and j are coordinates on the board, where (1,1) is the bottom left
$Rot(k)$: object k is the rotation of the piece
$Id(l)$: object l is the id of the piece
$F(i,j)$: This is true if tile (i,j) is filled by a unit of a tetromino

$E(i, j)$: This is true if tile (i,j) is empty and not filled by a unit of a tetromino

$C(i)$: This is true when row i is filled with pieces and is cleared

$O(i, j, k, l)$: This is true when an O piece is anchored at (i, j) with SRS state k and with id l (since more than one of the same piece can be used at once and id is used to separate individual pieces)

$I(i, j, k, l)$: This is true when an I piece is anchored at (i, j) with SRS state k and with id l (since more than one of the same piece can be used at once and id is used to separate individual pieces)

$L(i, j, k, l)$: This is true when an L piece is anchored at (i, j) with SRS state k and with id l

$J(i, j, k, l)$: This is true when an J piece is anchored at (i, j) with SRS state k and with id l

$S(i, j, k, l)$: This is true when an S piece is anchored at (i, j) with SRS state k and with id l

$Z(i, j, k, l)$: This is true when an Z piece is anchored at (i, j) with SRS state k and with id l

$T(i, j, k, l)$: This is true when an T piece is anchored at (i, j) with SRS state k and with id l

$P(i)$: Is true when a piece is type i

$N(i)$: The number of pieces active in i row

Converting Some of the Constraints:

$\forall i.(O(i, j, k, l) \wedge I(i, j, k, l) \wedge L(i, j, k, l) \wedge J(i, j, k, l) \wedge S(i, j, k, l) \wedge Z(i, j, k, l) \wedge T(i, j, k, l)) \rightarrow i \in 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$
All the pieces must have a row value that is within 1-10.

$\forall j.(O(i, j, k, l) \wedge I(i, j, k, l) \wedge L(i, j, k, l) \wedge J(i, j, k, l) \wedge S(i, j, k, l) \wedge Z(i, j, k, l) \wedge T(i, j, k, l)) \rightarrow j \in 1, 2, 3, 4$
All the pieces must have a column value that is within 1-4.

$\forall k.(O(i, j, k, l) \wedge I(i, j, k, l) \wedge L(i, j, k, l) \wedge J(i, j, k, l) \wedge S(i, j, k, l) \wedge Z(i, j, k, l) \wedge T(i, j, k, l)) \rightarrow k \in 1, 2, 3, 4$
All the pieces must have a rotation value that is within 1-4.

$\forall l.(O(i, j, k, l) \wedge I(i, j, k, l) \wedge L(i, j, k, l) \wedge J(i, j, k, l) \wedge S(i, j, k, l) \wedge Z(i, j, k, l) \wedge T(i, j, k, l)) \rightarrow l \in 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$
All the pieces must have a l value that is within 1-10.

$\forall i, j.F(i, j) \rightarrow \neg E(i, j)$ If the coordinate is filled, it can not be empty.

$I_x = 1, 2...10, I_y = 1, 2, 3, 4$ where $\forall x \in I_x, \forall y \in I_y$ this way x and y will act as coordinates for the board that we are working with which is shrunk down due to the limitations of the all clear with 10 tetrominos. This will then apply for all tetrominos.