Games file

Ts = time stamp

Event_type = summary of game events

SID = user id number

ECID = always is NIL

Session = date and time (yyyy-mm-dd_hr-mn-ss)

game_type = particular game experiment condition

game_number = they play as many games as possible in one hour. This indicates which game they played in order

episode_number = an episode begins when a game block appears at the top and ends when it lands at the bottom. This measures how many episodes happened.

level = highest level attained for that game

score = how many points earned

lines_cleared = how many lines they cleared while playing the game

completed = whether they completed that game (I think not sure)

game_duration = how long the game lasted in seconds

avg_ep_duration = average time player spends dealing with each game block

zoid_sequence = the order of the game blocks that spawned at the top and descended

Episodes file

The first few columns are the same as above. Beginning with the new ones for this file type below.

Episode = each game block that falls, Zoid = game block

curr_zoid = identity of the current game block for that episode

next_zoid = identity of the game block that will appear on the next episode

danger_mode = is the game in the top part of the screen when loss is imminent (and the music plays faster)

evt_sequence = what button was pressed and when

rots = number of game block rotations

trans = number of sideways translations

path_length = adding together rotations and translations min_rots = minimum rotations needed to put game block in place

min_trans = minimum translations needed to put game block in place

min_path = adding together min_rots and min_trans

min_rots_diff = difference between number of rotations needed to put game block in place and number of rotations actually made

min_trans_diff = difference between number of translations needed to put game block in place and number of rotations actually made

min_path_diff = difference between path_length and min_path

u_drops = “user drops” means how many units down the player intentionally descended the game block

s_drops = unknown and not used

prop_u_drops = “proportion of user drops” proportion of screen for which the game block is dropped, 1.00 would mean the whole screen

initial_lat = “initial latency” amount of time after the game block spawns when the player begins interacting with it

drop_lat = Drop latency. Time elapsed in milliseconds from the start of the episode until the player first drops the zoid.

avg_latency = Average latency. The mean time between all keypresses in an episode.

tetrises_game = how many tetrises (4-line clears) have been made in the game to that point

tetrises_level through zoid rep = unknown and not important

smi* and fix* = eye tracking data not currently implemented

all_diffs = differences between all column heights

all_ht = height of all columns added together

all_trans = unknown and not important

cd_1 = difference in height between columns 1 and 2

cd_2 = difference in height between columns 2 and 3

cd_3…..cd_9 = difference in height between columns 3 and 4….columns 9 and 10

cleared = lines cleared at the end of the episode

column_9 = height of the 9th column

cuml_cleared = unknown and not important

cuml_eroded = unknown and not important

eroded_cells = unknown and not important

full_cells = unknown and not important

max_diffs = unknown but could be figured out if necessary

max_ht_diff = unknown but could be figured out if necessary

mean_pit_depth = unknown but could be figured out if necessary

min_ht_diff = unknown but could be figured out if necessary

move_score = unknown but could be figured out if necessary

nine_filled = unknown but could be figured out if necessary

tetris_progress = unknown but could be figured out if necessary

tetris = whether or not a tetris was scored at the end of the episode

Pile orderliness: measures of the overall “randomness” or “order” of the pile.

jaggedness **-** Jaggedness. The perimeter of the top of the pile. A lower value implies a flatter pile, while a higher value implies a craggier surface.

col_trans, row_trans **-** Column and row transitions. Number of times a cell changes from open to closed along either columns or rows. This generally measures the “randomness” of the pile; a tall pile with no pits would rate low, as would a completely empty board, whereas a checkerboard pattern or completely random pile (i.e., riddled with pits and overhangs) would rate high.

pattern_div - Pattern diversity. This measure compares the pattern of empty and filled cells in each column, and the same for each row. Lower scores imply similar patterns across the pile, whereas a higher score implies more variability in the patterns created.

weighted_cells **-** Weighted cells. A count of the total number of filled cells in all columns, each weighted by its own height. The same number of total cells filled arranged flatly would rate lower than a those same cells stacked entirely along one wall, as more cells would be weighted higher due to their height.

Column heights: measures of the heights of each column of the game board.

mean_ht, max_ht, min_ht - Mean, Maximum, and Minimum height. The mean, maximum, and minimum height among all 10 columns in the pile. Note that only maximum height needs to exceed 20 to result in a game over.

d_all_ht = delta in all column heights. The change in all_ht after the game block lands into place and the episode ends.

d_mean_ht = delta in average column heights. The change in mean_ht after the game block lands into place and the episode ends.

Zoid-placement: local measures of the current game block’s position at the end of the episode.

landing_height - Landing height. The height of the bottom of the zoid’s final position.

matches - Matches. The number of edges of the zoid (in its final position) that border a filled cell. A low number implies precarious positioning of the zoid, whereas a higher number implies a zoid fitting more “snugly” into the surrounding pile.

d_max_ht - Delta maximum height. The change in the max_height score after placing the zoid and clearing any filled lines. A negative value implies line clears, while a zero-value implies the zoid was not placed at the top, and positive values imply the zoid pushed the top of the pile higher (and closer to failure).

d_pits - Delta pits. The change in the pits score after placing the zoid and clearing any filled lines. Negative values imply lines were cleared and pits were successfully opened up, while positive values imply the zoid placement created one or more new pits.

d_max_ht = delta in the maximum column heights. The change in max_ht after the game block lands into place and the episode ends.

Pits: the unworkable covered holes that prevent line-clears.

pits - Pits. The number of empty cells in the pile that are covered from above. As a player must fill an entire row to clear it, pits must have the cells above them cleared before their row can be cleared.

pit_depth - Pit depth. The sum of all pits weighted by the number of filled cells above them in a column. This score gives more weight to pits buried deeper in the pile.

pit_rows - Pit rows. The number of rows containing pits. This measure considers any number of pits in one row to be equivalent, as clearing everything above such a row would uncover all of its pits simultaneously.

lumped_pits - Lumped pits. A measure of pits considering all adjacent groups of pits to be identical. Thus four isolated pits in the pile would have more weight than one 2x2 cluster of pits.

Wells: the low-height columns surrounded by higher columns on either side which are relevant for fitting certain zoid shapes.

wells - Wells. The number of empty, uncovered cells with a filled cell on either side. The deeper a well is, the harder it is to work with. Yet deep wells are also associated with the highest scoring 4-line-clear Tetris maneuver.

deep_wells - Deep wells. The number of consecutive well segments of depth 3 or more. These are unique in the game in that they can only be filled by an I-zoid without creating one or more pits.

cuml_wells - Cumulative wells. Similar to wells, but weighing each segment of the well heavier as it goes deeper. A well of depth 1 evaluates as 1 (1); a well of depth 2 evaluates to 3 (2+1); a well of depth 3 evaluates to 6 (3+2+1); and so on.

max_well - Maximum well. The depth of the deepest well.