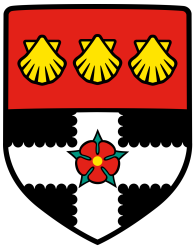

UNIVERSITY OF READING

DEPARTMENT OF COMPUTER SCIENCE

COMPUTER SCIENCE UNDERGRADUATE - RUBIK'S CUBE SOLVER LOGBOOK

CALLUM CLATON DAVID MCLENNAN

Supervisor: MARTIN LESTER

February 24, 2021

Rubik's Cube Solver - Logbook

Date	Description
10/08/2020	Created project. Followed tutorial to create emulator which allowed me to implement moves to the cube, animations when the moves occured and a scramble function but it doesn't work yet.
12/08/2020	Refind scrambling function, the cube now successfully scrambles with random moves.
14/08/2020	Reverse the Rubik's cube scramble to give the illusion that it's solved (Was apart of the tutorial). Updated the README with what some gifs and research information.
16/08/2020	Created multiple cube sizes. The only limitation on size is the computing power available. The program generates cubies within the cube which is unecessarily using up computing power. Moves/scrambling is only catered for 3x3x3 cube.
17/08/2020	Removed inner cubes, Set face colours to only the visible faces, Added FPS count, Added cube size, Added speed control, Fixed even numbered cube scramble. Catered moves to work for any sized cube Working on identifying when colours on neighbouring cubies faces match. I will then work on a human algorithm to solve the cube. After, I will begin working on a computer algorithm to solve the cube Besides these objectives, I'll be refining and adapting code to be easier to read and fit needs
30/08/2020	In process of adding X, Y, Z whole cube rotations. Algorithm files have been made, In progress with creating X,Y,Z rotations for ease of solve, Issues with setting cubies to correct positions Cubies change colours but they need to change positions on cube.
04/09/2020	Finished new turning functions Updated README. Fixed a few rotation/moving bugs. Created a few debugging print functions. Turning functions do not currently cater for any cube size that is not 3x3x3 **TODO**: Cater turning functions for cubes of any size, Adapt code to correctly act on double moves, Fix reverse scramble function, Clean code up for better understanding, Remove X,Y,Z rotations from counting as scramble/solve ,, moves since they don't modify the cube's scramble state Start working on human solving algorithm after these jobs are complete.

Date	Description
09/09/2020	Catered the turning functions for all cube sizes. Added more control over cubies Refined controls over speed Can switch between cube sizes during runtime Cleaned up some code **TODO**: Fix cubie colour change problem that occurs for unknown reason Clean up code to be more conventional. Upon further research, I've realised I probably went a bit overkill with the comments
12/09/2020	Updates to human algorithm function - Solves white cross on cube. Currently working on additional steps of the human algorithm. Although it's not a mandatory requirement, I feel it will heavily familiarise me with every aspect of the cube - ready for more complex algorithms. Added some boolean operators for when the program should call the solving function Also when to hide the HUD - gives the program a cleaner look Issues - Going to disregard issues for higher cubes until I've finished the human algorithm catering for a 3x3x3 (I'm still unsure the reason for random colour changes for bigger cubes as of yet.) **TODO**: - Add 2D visualisation of the cube - Allow user to custom scramble the cube by clicking 2D visualisation - Research and discover methods of speeding up the programs performance as cube's above 25x25x25 puts FPS below 30.
13/09/2020	Finished stage 2 of human algorithm Successfully arranges white cross on cube Successfully positions corners on bottom of cube - Changed README banner
15/09/2020	Completed step 3 (F2L) on human algorithm - Discovered looping issue with step 2 - now fixed. - Still need to clean up code after all steps are completed for algorithm - Progressing onto step 4 - Updated README
17/09/2020	Finished human algorithm steps 3,4,5,6 and 7. Successfully solves the Rubik's cube. Prints most of the solving moves to HUD Need to refine code to make easier to understand. Need to clean code up Planning on adding 2D view of the Rubik's RubiksCube Going to do some research of local search algorithm Will postpone catering for bigger cubes until further notice.
19/09/2020	Added 2D visualisation of cube Hoping to allow the user to modify the cubie's colours via the 2D visualisation in the near future Cleaned up code (re-evaluated comments)

Date	Description
28/10/2020	On the hunt for what's causing the cube to randomly change scramble states. I suspect it could be applying the "solution" it comes up with straight away then applying it again with animations? I'm unsure as of yet as further research is required. I'm also working on trying to clean up the code to make it more logical and easy to follow.
12/11/2020	Spending time reading up and learning more on preexisting solving algorithms. Have discovered a cheaper cube object would be good for me to implement - computationally cheaper object will result in faster solving speeds. Trying to get some coursework done so there may be less frequent updates.
28/11/2020	Have tried refining code but not much progress has been made here as I've been focusing on coursework!
15/12/2020	Have added and continuing to work on a 'cheaper' cube object that requires less computational power to work with when trying to solve the cube.
27/12/2020	Attempting to implement methods used by the Kociemba's algorithm to better learn what's required for solving the cube. Have learnt the hard way that straight up brute forcing solutions less than 7 moves isn't a very reasonable tactic.
10/01/2021	Have backtracked a bit to solidify understanding of what's required to best solve a cube. Thanks to online resources, I've managed to implement an IDDFS solver for a 2x2 cube. This requires a specific cubie to be oriented in a specific way. Hopefully this acts as a stepping stone to more meaningful solver implementations in the future! I've also decided my local search approach may be losing it's precedence as I learn more about alternative solving methods.
18/01/2021	Read up on Korf's Algorithm which seems to be the most efficient solving algorithm in regards to 20 move or less solves with the only downside being time taken. This involves "pruning tables" which is a new area/concept to me but this seems to cross over to older solving algorithms so implementing this could make it easier for me to understand and implementing Thistlethwaite's and Kociemba's solving algorithms - more "optimal" algorithms I've been impressed by.
07/02/2021	After too much testing and lots of research, I've managed to implement a working Korf's Algorithm for solving my cube. During this process, I came across the "Reduction method" which is the stepping stone that could mean achieving my earlier amibitious goal of solving larger cubes... We shall see! Currently working on adding Thistlethwaite's although it seems it will require some additional pruning tables than were used with Korf's (Pruning tables are the secret sauce to what make these solving algorithms work so well).
20/02/2020	Have implemented Thistlethwaite's algorithm but still working on some additional pruning tables to speed up the search for solutions in G2 and G3 groups. Have discovered a method of adding GUI elements via ControlP5 library. Will be focusing on the UI side of things as I've mostly been working on the functionality. I feel I'm at a sufficient level of progress to start catering it for public use.