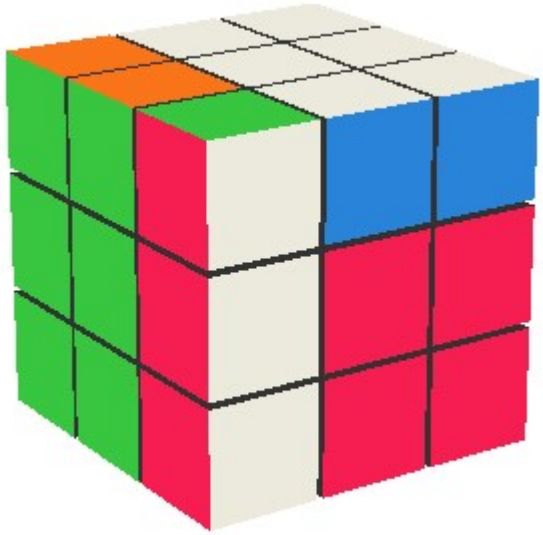
יריב גדידי – הקוביה ההונגרית

תוכן עניינים

כללי..... 1
ארכיטקטורה..... 2
מודל הקובייה ה"תלת מימדי"..... 4
מודל הקוביה השטוח..... 5
סיבוב פאה במודל התלת מימדי..... 6
פתרון הקוביה ע"י אלגוריתם של "מכונת מצבים": 7



כללי

הפרוייקט נבנה כפרוייקט אישי לימודי שלי (לימוד עצמי). האפליקציה מאפשרת להציג קוביה הונגרית, לערבב ולפתור, וכן לסדר את הקוביה הוירטואלית לפי סידור של קוביה פיזית כאשר האפליקציה תדע לפתור את הקוביה.

בעת הסידור:

- האפליקציה לא מאפשרת סידורים לא חוקיים כגון פינה (אדום, כתום, לבן)
- האפליקציה יודעת להשלים פינות ע"פ אלימינציה של חוקיות החלקים.
- האפליקציה **לא** תדע לזהות סידור לא חוקי, כזה שהיה מתקבל לו היינו מפרקים פיזית קוביה ומרכיבים אותה לא נכון (הדבר אפשרי ע"י זיהוי קוביה לא פתירה, אבל לא מומש).

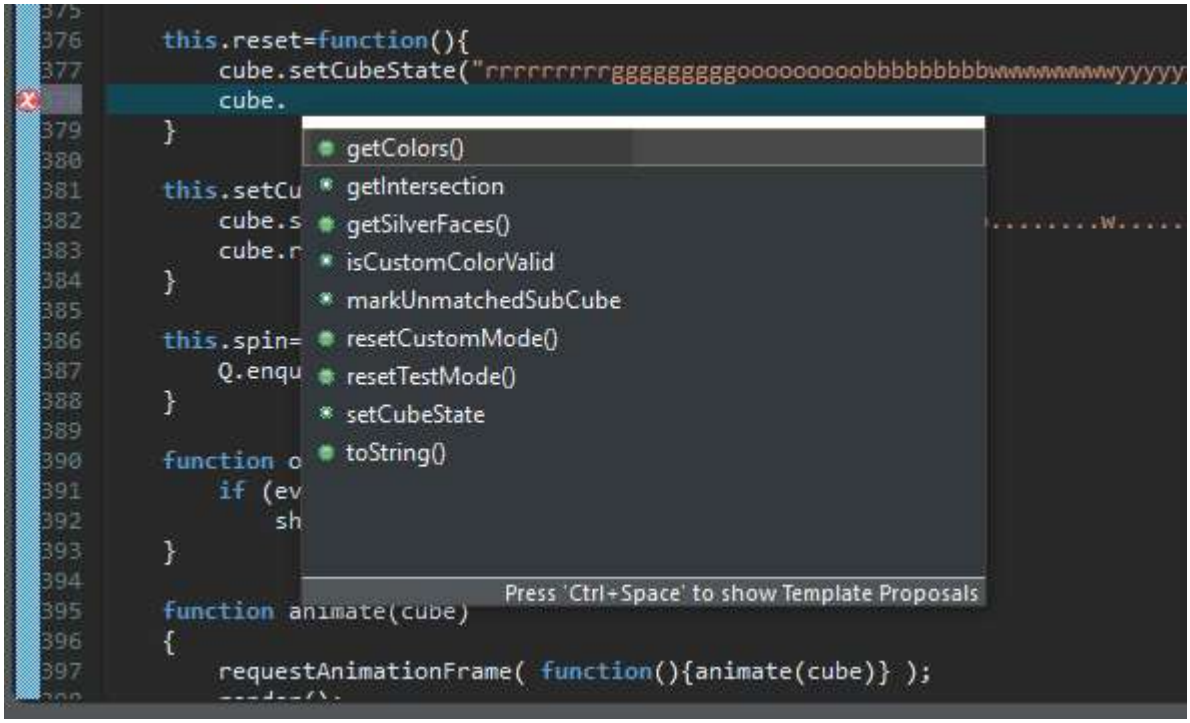
ארכיטקטורה

הפרוייקט בוצע ב javascript, שימוש ב Three.JS framework לצורך הצגת המודל וב AngularJS לממשק המשתמש.

מבנה היררכי:

אובייקט RubiksCube:

- ממדל את הקובייה כמערך של "תת קוביות" כלומר חלקי הקוביה כאובייקטי BoxGeometry של TheeJS
- חושף פונקציות כגון ()toString אשר נותן יצוג של הקוביה כאוסף תווים (יוסבר בהמשך) ו – ()setCubeState המאפשר לסדר את הקובייה לפי סידור ראשוני מקלט של string. וכמובן ()rotate המאפשר סיבוב פאה.
- מנהל את ה transformatios השונים של סיבוב הקובייה.



אובייקט RubiksView

- מנהל את ה"זירה" (THREE.Scene())
- מבצע את "הלולאה הראשית" הדרושה להמשכיות של התוכנית והאנימציה
- מיישם תור של פעולות, לדוגמה דחיפה של `'u', 'l!', 'u!'` לתור יבצע סיבוב של פאה עליונה, שמאלית הפוכה (נגד כיוון השעון) ועליונה הפוכה
- מיישם "מכונת מצבים" שהיא הבסיס לפיתרון לדוגמא: `'u', 'l!', 'u!', 'n', 4` כאשר הרצף n,4 נמשך מן התור, הוא מורה לקוביה לעבור למצב 4
- חושף לשכבה העליונה (לאפליקציה) פונקציות כגון solve(), scrumble() ו setClickStateColor() אשר מורה לקוביה: "מרגע זה, כל נגיעה תגרום לפאה בה נגעו להיצבע בצבע זה".

rubiks-app

אפליקציית אנגולר אשר מתווכת בין כפתורי ה HTML לממשק ה VIEW

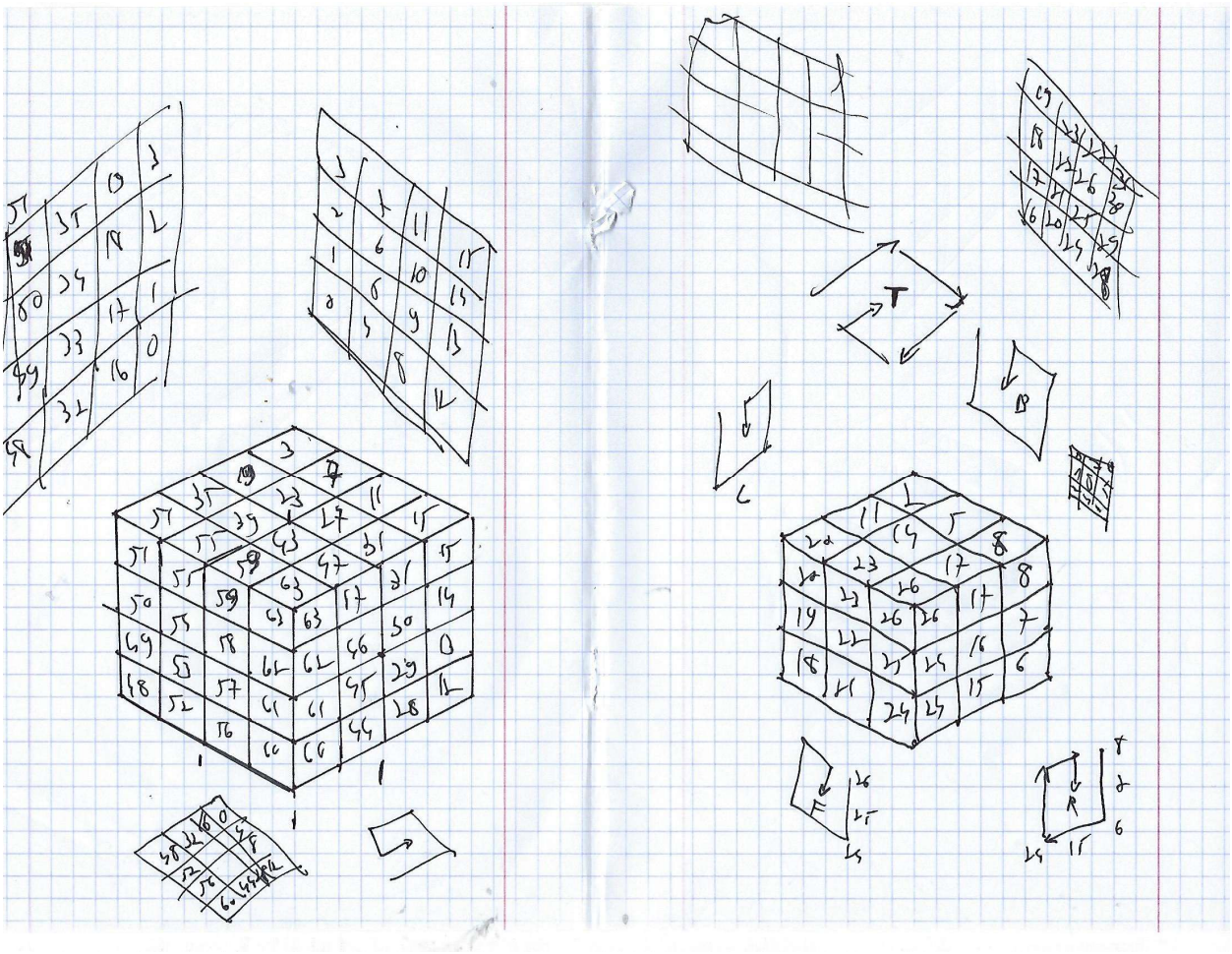
האפליקציה היא בעלת "מצבים":

- Idle
- Scrumble
- Solve
- Custom – מצב זה, משנה את ה UI מכפתורי "סיבוב" לכפתורי "בחירת צבע" ומאפשר למשתמש "לתכנת" את הקוביה למצב רצוי.

מודל הקובייה ה"תלת מימדי"

תת קוביה היא אובייקט box בסיסי של ThreeJS framework
המודל התלת מימדי מגדיר את הקוביה ההונגרית כאוסף של "תת קוביות"

- Cube הינו מערך המכיל את כל תת הקוביות (27 במקרה של 3*3 למרות שתת קוביה האמצעית לא משתתפת)
- Faces הינו רשימה של פאות כאשר כל פאה היא רשימה של ה תת קוביות המרכיבות אותה
- rotationMatrix – אובייקט אשר מגדיר את ההחלפות בין הפאות במקרה של סיבוב – יוסבר בנפרד
- הסידור של הקוביות בתוך הפאות נבחר להיות בצורת חילזון: לדוגמה פאה front תחיל את תת קוביות 22 ,23 ,20 ,19 ,18 ,21 ,24 ,26 על פי הסדר הזה
  העקביות הזו מאפשרת לנהל את "ההחלפות" בצורה יעילה ע"י rotationMatrix

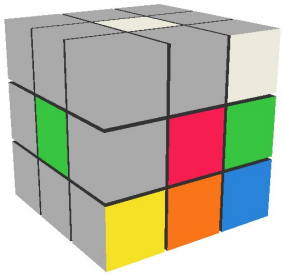


מודל הקוביה השטוח

המודל השטוח מאפשר "לסרלז" (serialize) את הקוביה למחרוזת תווים, הדבר נחוץ לצורך שימוש במכונת המצבים, כפי שיוסבר בהמשך

מכיוון שהמודל קל להבנה אבל קשה להסבר אשתמש בדוגמה:

w = white, g = green, b = blue, o = orange, y = yellow, r = red
'.' = gray == Don't care
wgboy...r.....g.....o.....b.....w.....y

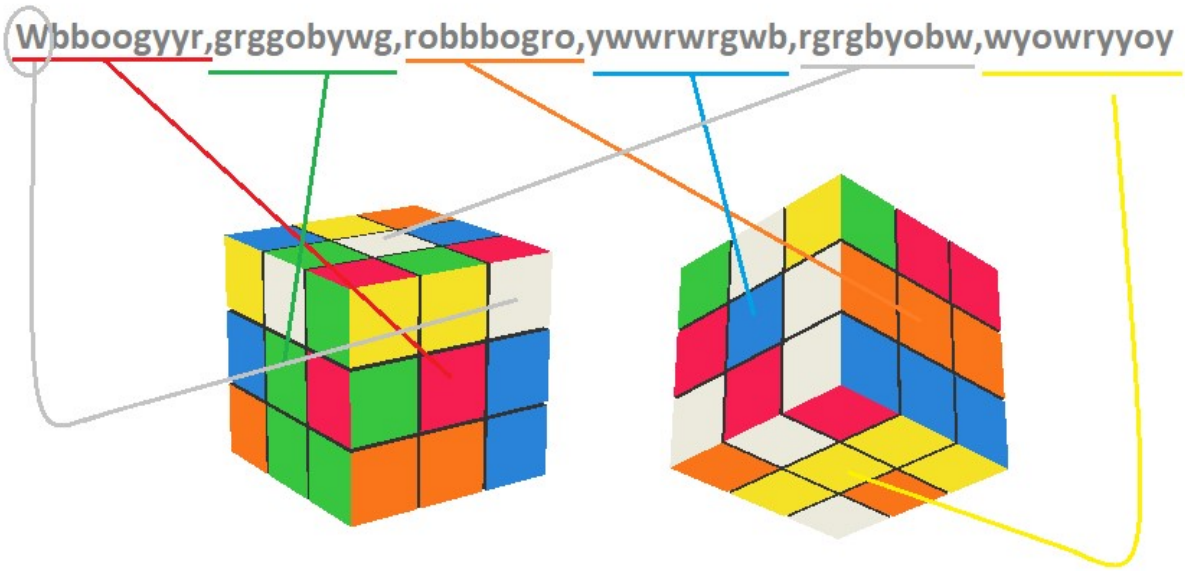


מתחילים בפינה ימנית עליונה של כל פאה (w – בדוגמה) ומתקדמים ספירלית כלפי מטה, שמאלה.. האות התשיעית בכל פאה מייצגת את צבע הפאה (קבוע – לא משתנה)

כשמסתיימת פאה, עוברים לפאה הבאה לפי סדר הפאות (נבחר שרירותית) : r, g, o, b, w , y

דוגמה נוספת:

Wbboogyyrgrggobywgrobbbogroywwrwrgwbrgrgbyobwwyowryyoy



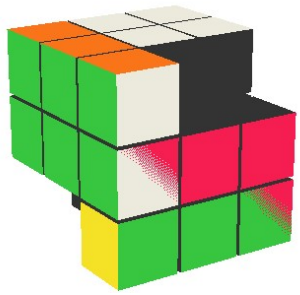
סיבוב פאה במודל התלת מימדי

הפעולה של החלת טרנספורמציה גאומטרית במרחב הינה חלק מארגז הכלים של ThreeJS Framework
השורות הבאות גורמות ל 9 תת קוביות השייכות לפאה להסתובב סביב ציר X בכיוון ובשיעור שנקבעו.
מבחינה חזותית האפקט מושג

```
matrix = new THREE.Matrix4().makeRotationX( direction * rotationFactor );

for (var i=0; i<9; i++)
    cube[face[i]].applyMatrix(matrix);
```

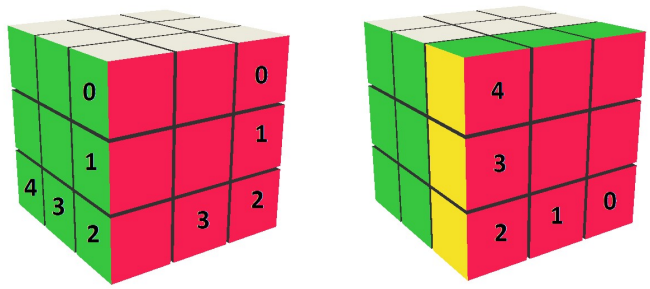
הבעיה שהסיבוב "שובר" את המבנה הלוגי, כך שבסיבוב הבא, לא נדע על אילו תת קוביות להכיל את הטרנספורמציה, והמבנה הכללי של הקוביה יהרס.



כדי להתגבר על כך, יש לבצע עדכון של המודל הלוגי:

בדוגמה הבאה סובבנו את פאה R:

- הקוביות ששייכות לפאה R לא השתנו (הם זזו אבל לא התחלפו)
- גם פאה O (הכתומה מאחור) לא הושפעה
- פאות G, W, B ו Y הושפעו באופן הבא:



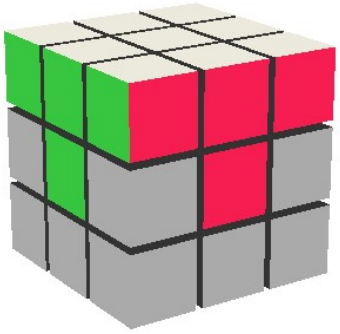
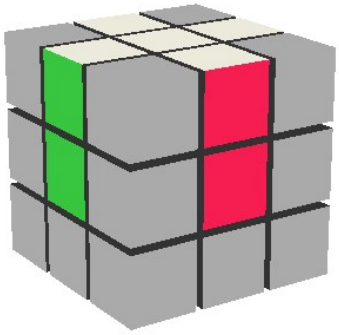
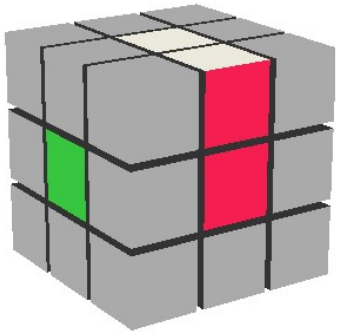
לאחר הסיבוב:

1. G[0]=R[4]
2. G[1]=R[3]
3. G[2]=R[2]

הטרנספורמציה של פאה G היא: 0,4,1,3,2,2 ובאופן דומה, מגדירים את יתר הטרנספורמציות בטבלה הבאה:

```
var rotationMatrix={
        // order = right front left back up down
        rightClockWise:{m:[null,[0,4,1,3,2,2],null,[4,0,5,7,6,6],[0,6,1,5,2,4],[0,2,1,1,2,0] ], targetFace:rightFace, isCounterClockwise: false},
        rightCounterClockWise:{m:[null,[0,0,1,7,2,6],null,[4,4,5,3,6,2],[0,2,1,1,2,0],[0,6,1,5,2,4] ],  targetFace:rightFace, isCounterClockwise: true} ,
        leftClockWise:{m:[null,[4,0,5,7,6,6],null,[0,4,1,3,2,2],[4,6,5,5,6,4],[4,2,5,1,6,0] ],  targetFace:leftFace, isCounterClockwise: false} ,
        leftCounterClockWise:{m:[null,[4,4,5,3,6,2],null,[0,0,1,7,2,6],[4,2,5,1,6,0],[4,6,5,5,6,4] ],  targetFace:leftFace, isCounterClockwise: true} ,
        frontClockWise:{m:[[6,6,5,7,4,0],null,[0,4,1,3,2,2],null,[2,6,3,5,4,4],[0,0,7,1,6,2] ], targetFace:frontFace, isCounterClockwise: false},
        frontCounterClockWise:{m:[[6,2,5,3,4,4],null,[0,0,1,7,2,6],null,[2,2,3,1,4,0],[0,4,7,5,6,6] ],  targetFace:frontFace, isCounterClockwise: true} ,
        backClockWise:{m:[[0,4,1,3,2,2],null,[6,6,5,7,4,0],null,[0,4,7,5,6,6],[2,2,3,1,4,0] ],  targetFace:backFace, isCounterClockwise: false} ,
        backCounterClockWise:{m:[[0,0,1,7,2,6],null,[6,2,5,3,4,4],null,[0,0,7,1,6,2],[2,6,3,5,4,4] ],  targetFace:backFace, isCounterClockwise: true} ,
        topClockWise:{m:[[0,6,7,7,6,0],[0,0,7,1,6,2],[0,2,7,3,6,4],[0,4,7,5,6,6] ,null, null],  targetFace:topFace, isCounterClockwise: false} ,
        topCounterClockWise:{m:[[0,2,7,3,6,4],[0,4,7,5,6,6],[0,6,7,7,6,0],[0,0,7,1,6,2] ,null,null],  targetFace:topFace, isCounterClockwise: true} ,
        bottomClockWise:{m:[[4,6,3,7,2,0],[4,4,3,5,2,6],[2,4,3,3,4,2],[2,2,3,1,4,0],null, null ],  targetFace:bottomFace, isCounterClockwise: false} ,
        bottomCounterClockWise:{m:[[4,2,3,3,2,4],[4,0,3,1,2,2],[2,0,3,7,4,6],[2,6,3,5,4,4],null,null ],  targetFace:bottomFace, isCounterClockwise: true} ,
};
```

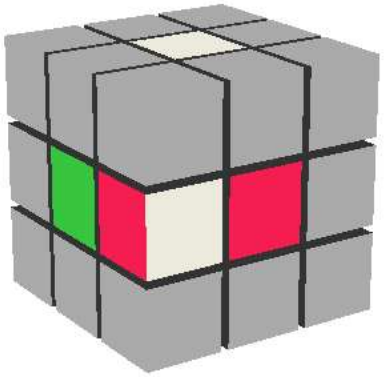
פתרון הקוביה ע"י אלגוריתם של "מכונת מצבים":

האלגוריתם שמימשתי הוא הפשוט ביותר (והאיטי ביותר). כפי שיוסבר, פתרון הקוביה לא קשור לקוד אלא לתכנות מכונת המצבים.

מהתבוננות בקוביה, כדי להביא את "אדום-לבן" למקומו, יש 24 מצבים (מיקומים) אפשריים. עבור כל מיקום כזה יש רצף (למעשה יותר מאחד) של תנועות שיעשו את העבודה.

לדוגמה:



```
{
    state: 0,
    initialState:'..............r....w.................................',
    moves: ['f','ul', 'n' , 1],
    finalState: 0}
},
```

לאחר מכן, ל"אדום-כתום" נותרו 22 מצבים אפשריים וכן הלאה, סה"כ כ 250 מצבים אפשריים לפתרון הקובייה כולה

ע"י סריאליזציה של הקוביה, באופן שהוסבר קודם, ניתן לאתר את המצב הנתון תוך שימוש ב regex, כאשר '.' (נקודה) מייצגת "צבע אפור" (שאינו קיים בפועל) בקוביה, וב regex "כל תיו" או don't care.

ניתן לראות באופן ויזואלי איך תוך התקדמות עם המצבים, הקוביה "הולכת ונפתרת"

```
{state: 0, initialState:'.................................r........w..........', moves: ['u', 'n' , 1], finalState: 0},
{state: 0, initialState:'.......................w....r.........................', moves: ['b!', 'u', 'n' , 1], finalState: 0},
{state: 0, initialState:'...............................r................w.....', moves: ['d!','r','r', 'n' , 1], finalState: 0},
{state: 0, initialState:'.w.............................r...................', moves: ['b','u', 'n' , 1], finalState: 0},

{state: 0, initialState:'...............w.....................r.............', moves: ['f', 'r', 'n' , 1], finalState: 0},
{state: 0, initialState:'.........................w..............r...........', moves: ['u!', 'f', 'r', 'n' , 1], finalState: 0},
{state: 0, initialState:'...............................w........r..........', moves: ['b!', 'r!', 'n' , 1], finalState: 0},
{state: 0, initialState:'.......w..........................r................', moves: ['u!', 'b!', 'r!', 'n' , 1], finalState: 0},

{state: 0, initialState:'............w.....................................r.', moves: ['f!', 'r', 'n' , 1], finalState: 0},
{state: 0, initialState:'...w.......................................r.......', moves: ['d!' ,'f!', 'r', 'n' , 1], finalState: 0},
{state: 0, initialState:'..........................w................r.....', moves: ['b', 'r!', 'n' , 1], finalState: 0},
{state: 0, initialState:'....................w...........................r...', moves: ['d','f!', 'r', 'n' , 1], finalState: 0},

{state: 1, initialState:'.......r................o...........w...w............', moves: ['n' , 2], finalState: 24 ,  stateName: 'white-orange'},
{state: 1, initialState:'.......r................w...........w...o............', moves: ['u!', 'f!', 'u' ,'l!', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r........o...................w.w..............', moves: ['u', 'l', 'u!', 'l!', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r.........................o..w.....w..........', moves: ['u!', 'l', 'u', 'l!', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r........w...................w.o..............', moves: ['f!', 'l!', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r.........................w..w.....o..........', moves: ['b', 'l', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r.............o...............w...........w...', moves: ['l', 'l', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r......................o......w..........w.....', moves: ['d', 'l', 'l', 'n' , 2], finalState: 24},
{state: 1, initialState:'...o...r...............................w........w.......', moves: ['d', 'd', 'l', 'l', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r.............w...............w............o...', moves: ['d', 'f', 'l!', 'n' , 2], finalState: 24},
{state: 1, initialState:'.......r....w.                        w               o ', moves: ['f', 'l!', 'n' , 2], finalState: 24}
```

```
{state: 6, initialState:'......rrrg...b.gggo.o....oo.......bb.wwwww.ww......w.y', moves:  ["d!", "b", "d!", "b!", "l!", "d!", "d!", "l", "n", 7], finalState:129}, //144
{state: 6, initialState:'......rrrg...w.gggo.b....oo.......bb.wwwww.ww......o.y', moves:  ["b", "d!", "b!", "n", 7], finalState:129},
{state: 6, initialState:'......rrrg...o.gggo.w....oo.......bb.wwwww.ww......b.y', moves: ["d", "l!", "d", "d", "l", "n", 7], finalState:129},

{state: 7, initialState:'r.....rrrg.....gggo.....ooob.....bbbwwwwwwwww........y', moves: [ "n", 8], finalState:147, stateName: 'white-red-blue'},
{state: 7, initialState:'b.....rrrg.....gggo.....ooob.....wbbrwwwwwwww........y', moves: ["b!", "d!", "b", "d", "d", "r", "d!", "r!", "n", 8], finalState:147},
{state: 7, initialState:'w.....rrrg.....gggo.....ooob.....rbbbwwwwwwww........y', moves: ["r", "d", "r!", "d!", "r", "d", "r!", "n", 8], finalState:147},

{state: 7, initialState:'....r.rrrg.b...gggo.....ooob......bb.wwwwwwwww.......y', moves:  ["d", "b!", "d", "b", "r", "d", "d", "r!", "n", 8], finalState:147}, //150
{state: 7, initialState:'....b.rrrg.w...gggo.....ooob......bb.wwwwwwwwr.......y', moves:  ["b!", "d", "b", "n", 8], finalState:147},
{state: 7, initialState:'....w.rrrg.r...gggo.....ooob......bb.wwwwwwwwb.......y', moves:  ["d", "b!", "d!", "b", "n", 8], finalState:147},

{state: 7, initialState:'..b...rrrg.....gggo.....ooob...r..bb.wwwwwwww..w.....y', moves:  ["b!", "d", "b", "r", "d", "d", "r!", "n", 8], finalState:147}, //153
{state: 7, initialState:'..w...rrrg.....gggo.....ooob...b..bb.wwwwwwww..r.....y', moves: ["r", "d", "r!", "n", 8], finalState:147},
{state: 7, initialState:'..r...rrrg.....gggo.....ooob...w..bb.wwwwwwww..b.....y', moves: ["b!", "d!", "b", "n", 8], finalState:147},

{state: 7, initialState:'......rrrg.....gggo...r.ooob.b....bb.wwwwwwww....w...y', moves: ["d!", "r", "d!", "r!", "b!", "d", "d", "b", "n", 8], finalState:147}, //156
{state: 7, initialState:'......rrrg.....gggo...w.ooob.r....bb.wwwwwwww....b...y', moves: ["r", "d!", "r!", "n", 8], finalState:147},
```

```
{state: 14, initialState:'rr..rrrrrggg..gggoo...oooobb[^b]..bbbbwwwwwwwwwyyyyyyyyy', moves: ['r!', 'b', 'r!', 'f', 'f', 'r', 'b!', 'r!', 'f','f', 'r','r','n',14], finalState:236, stateName: 'yelow'},
{state: 14, initialState:'rr..rrrrrggg..gggoo...oooobb..[^b]bbbbwwwwwwwwwyyyyyyyyy', moves: ['r!', 'b', 'r!', 'f', 'f', 'r', 'b!', 'r!', 'f','f', 'r','r','n',14], finalState:236, stateName: 'yelow'},
{state: 15, initialState:'rrrrrrrrrggggggggoooooooobbbbbbbbbwwwwwwwwwyyyyyyyyy', moves: ['n',-1], finalState:241, stateName: 'yelow'}, //241
{state: 15, initialState:'rrr[^r]rrrrrggg[^g]gggggooo[^o]ooooobbb[^b]bbbbbwwwwwwwwwyyyyyyyyy', moves: ['f', 'f', 'd', 'l!', 'r', 'f','f', 'l', 'r!', 'd', 'f','f', 'n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrrrrrrrrggg.gggggooooooobbb.bbbbbwwwwwwwwwyyyyyyyyy', moves: ['l','l', 'd!', 'b!', 'f', 'l', 'l', 'b', 'f!', 'd!', 'l', 'l','n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrrrrrrrrggg.gggggooogoooooobbb.bbbbbwwwwwwwwwyyyyyyyyy', moves: ['l','l', 'd', 'b!', 'f', 'l', 'l', 'b', 'f!', 'd', 'l', 'l','n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrr.rrrrrggggggggoooo.oooooobbbrbbbbbwwwwwwwwwyyyyyyyyy', moves: ['b','b', 'd!', 'r!', 'l', 'b', 'b', 'r', 'l!', 'd!', 'b', 'b','n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrr.rrrrrggggggggoooo.oooooobbbrbbbbbwwwwwwwwwyyyyyyyyy', moves: ['b','b', 'd', 'r!', 'l', 'b', 'b', 'r', 'l!', 'd', 'b', 'b','n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrrgrrrrrggg.ggggggoooooooobbb.bbbbbwwwwwwwwwyyyyyyyyy', moves: ['r','r', 'd', 'f!', 'b', 'r', 'r', 'f', 'b!', 'd!', 'r', 'r','n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrrbrrrrrggg.ggggggoooooooobbb.bbbbbwwwwwwwwwyyyyyyyyy', moves: ['r','r', 'd', 'f!', 'b', 'r', 'r', 'f', 'b!', 'd', 'r', 'r','n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrr.rrrrrgggggggooo.oooooobbbbbbbbbwwwwwwwwwyyyyyyyyy', moves: ['f','f', 'd!', 'l!', 'r', 'f', 'f', 'l', 'r!', 'd!', 'f', 'f','n',15], finalState:241, stateName: 'yelow'},
{state: 15, initialState:'rrr.rrrrrgggrgggggooo.oooooobbbbbbbbbwwwwwwwwwyyyyyyyyy', moves: ['f','f', 'd', 'l!', 'r', 'f', 'f', 'l', 'r!', 'd', 'f', 'f','n',15], finalState:241, stateName: 'yelow'},

]:
```