Daniel Bermudez Galvan 1492657

TQS_SIMON

Funcionalitat:

- Initialize()

-Inicialitzar els valors de la pantalla Score a 0 i Level 1 en cas que no estiguin per defecte.

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

-Caixa Blanca / test unitari

-El test comprova si s'ha inicialitzat amb els valors Score = 0, Level = 1

Retorna True o False

```
describe('Game-Initialize', () => {
    it('Comprobar que al inicializar los valores Score = 0 y Nivel = 1 esten correctamente inicializados', () => {
        const score_nivel = game.initialize();
        var InitCorrecte = false
        if(score_nivel[0] == 0 && score_nivel[1] == 1)
        {
            InitCorrecte = true;
        }
        expect(InitCorrecte).toBe(true);

    });
})
```

Funcionalitat:

-Array()

-Generar un array de per exemple tamany 10, en que els valors de l'array nomes poden anar del 0-3 (Color verd, blau, Vermell i groc) (He ficat la condició, que el tamany del array nomes pot anar del 1-10.)

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

-Caixa Blanca/ Test unitari

-Faig la comprovació dels 4 casos possibles, que le diguin que el tamany ha de ser del:

"1-10 (Correcte) , negatiu (Incorrecte), 0(incorrecte), <10(Incorrecte)"

I que en els casos que dona incorrecte li doni un valor per defecte:

>10 = 10 // 0 o <0 = 1

```
describe('Game-Array', () => {

    it('Comprobar en caso que la array maxima sea de 10 si se genera un array de 10 elementos', () => {
        const resultado = game.array(10);
        expect(resultado[0].length).toBe(10);

    });
    //Fallo porque pasen el 0
    it('Comprobar si al pasarle a array un 0 devuelve una cadena de tamaño 1 por defecto', () => {
        const resultado = game.array(0);
        expect(resultado[0].length).toBe(1);
    });

    //Fallo porque sea negativo
    it('Comprobar si al pasarle a array un valor negativo devuelve una cadena de tamaño 1 por defecto', () => {
        const resultado= game.array(-5);
        expect(resultado[0].length).toBe(1);
    });

    //Fallo porque sea mas grande que 10
    it('Comprobar en caso que la array maxima sea mayor de 10 si se genera un array de 10 elementos por defecto', () => {
        const resultado = game.array(16);
        expect(resultado[0].length).toBe(10);
    });
```

Test:

-Arxiu: IndiceJS.js

-Class: Game

-Caixa Blanca/ Test Condition

- Comprovo que :

El numero que em donen per crear l'array si es 0 o no, i retorno un boolean.

El numero que em donen per crear l'array es negatiu, i retorno un boolean.

```
//1 Condition
it('Comprobar Condicion 1 LN == 0 que sea true', () => {
    const resultado = game.array(0);
    expect(resultado[2]).toBe(true);
});

it('Comprobar Condicion 1 LN == 0 que sea false', () => {
    const resultado = game.array(6);
    expect(resultado[2]).toBe(false);
});
//2 Condition
it('Comprobar condicion 2 LN = Numero negativo sea true', () => {
    const resultado= game.array(-5);
    expect(resultado[3]).toBe(true);
});
it('Comprobar condicion 2 LN = Numero negativo sea false', () => {
    const resultado= game.array(6);
    expect(resultado[3]).toBe(false);
});
```

Test:

-Arxiu: IndiceJS.js

-Class: Game

-Caixa Blanca / Test Decision

-Comprovo que:

El numero que em donen per crear l'array si es 0 o es negatiu i retorno boolean.

Comprovant així que puc obtenir en aquest cas, els casos

True-False // False-True // False-False

```
//1+2 Decision
it('Comprobar combinaciones posibles de la condicion 1 + 2 en este caso True-False', () => {
    const resultado = game.array(0);
    expect(resultado[1]).toBe(true);
});

it('Comprobar combinaciones posibles de la condicion 1 + 2 en este caso False-True', () => {
    const resultado= game.array(-5);
    expect(resultado[1]).toBe(true);
});

it('Comprobar combinaciones posibles de la condicion 1 + 2 en este caso False-False', () => {
    const resultado= game.array(6);
    expect(resultado[1]).toBe(false);
});
```

Funcionalitat:

-TrasnformNumberToColor

-Transforma el numero que li donen a un dels 4 colors disponibles (Verd, Blau, Vermell o Groc) segons el numero que sigui.

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

-Caixa Negra / Valors Limits-Frontera

- Primer comprovo els valors frontera (0,3), despres comprovo el casos intermitjos (1,2), el límits que sèrie (-1 i 1) per part del 0, i (2,4) per part del 3, i ja per finalitzar comprovo valors molt superiors e inferiors a aquests com 20 i -20, retornen en cadascun del casos el color o "Error".

Test:

-Arxiu: IndiceJS.js

-Class: Game

-Caixa Negra / Partició

- Comprovo els valors que son Vàlids i els que no son vàlids, deixant com a vàlids

Del 0-3 i com a invàlids tots els altres <0 i >3.

```
describe('Game-TransformNumberToColor', () => {

    //Valores Validos
    //Fronteras  (0-3)
    it('Comprobar en caso que le llegue el numero 0 del array devuelve el color Verde', () => {
        const color = game.transformNumberToColor(0)
        expect(color).toBe('Verde');
    });
    it('Comprobar en caso que le llegue el numero 3 del array devuelve el color Amarillo', () => {
        const color = game.transformNumberToColor(3)
        expect(color).toBe('Amarillo');
    });

    //Casos intermedios
    it('Comprobar en caso que le llegue el numero 1 del array devuelve el color Azul', () => {
        const color = game.transformNumberToColor(1)
        expect(color).toBe('Azul');
    });
    it('Comprobar en caso que le llegue el numero 2 del array devuelve el color Rojo', () => {
        const color = game.transformNumberToColor(2)
        expect(color).toBe('Rojo');
    });

    //Valores Invalidos
    //Limite 0  (-1, 1)
    it('Limite de 0 por abajo es el -1 (el 1 ya esta comprobado en casos intermedios)', () => {
        const color = game.transformNumberToColor(-1)
        expect(color).toBe('Error');
    });
    //Limite 3 (2,4)
    it('Limite de 3 por arriba es el 4 (el 2 ya esta comprobado en casos intermedios)', () => {
        const color = game.transformNumberToColor(4)
        expect(color).toBe('Error');
    });
    //Valores muy superiores/inferiores
    it('Valores muy superiores como 20', () => {
        const color = game.transformNumberToColor(20)
        expect(color).toBe('Error');
    });
    it('Valores muy inferiores como -20', () => {
        const color = game.transformNumberToColor(-20)
        expect(color).toBe('Error');
    });
```

Funcionalitat:

-TransformColorToNumber

-Transforma un dels 4 Colors a un numero del 0-3 (el que li representa) això ens serveix per quan fas click en la pantalla en algun color, detecta aquest color, i et retorna el color, i es comprova amb el array comentat abans, per veure si es correcte o no el seleccionat.

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

- Caixa Negra / Partició

- Comprovo els valors que tenen solució numèrica del 0-3 es a dir que son vàlids, que son el color Verd, Blau, Vermell i Groc, en canvi si arriba un altre color, numero, caràcter o inclús si es un color correcta però no esta en el format “Amarillo” sinó que esta “amarillo” o “aMarillo” donarà un missatge de Error.

```
describe('Game-TransformColorToNumber', () => {

    //Valores Validos
    it('Comprobar en caso que le llegue el color Verde devuelve el valor 0', () => {
        const color = game.transformColorToNumber('Verde')
        expect(color).toBe(0);
    });
    it('Comprobar en caso que le llegue el color Azul devuelve el valor 1', () => {
        const color = game.transformColorToNumber('Azul')
        expect(color).toBe(1);
    });
    it('Comprobar en caso que le llegue el color Rojo  devuelve el valor 2', () => {
        const color = game.transformColorToNumber('Rojo')
        expect(color).toBe(2);
    });
    it('Comprobar en caso que le llegue el color Amarillo devuelve el valor 3', () => {
        const color = game.transformColorToNumber('Amarillo')
        expect(color).toBe(3);
    });
    //Valores invalidos
    it('Comprobar en caso que le llegue otro color devuelva error', () => {
        const color = game.transformColorToNumber('Rosa')
        expect(color).toBe('Error');
    });
    it('Comprobar en caso que le llegue otro color devuelva error', () => {
        const color = game.transformColorToNumber(0)
        expect(color).toBe('Error');
    });
    it('Comprobar en caso que le llegue otro color devuelva error', () => {
        const color = game.transformColorToNumber('!')
        expect(color).toBe('Error');
    });
    it('Comprobar en caso que le llegue otro color devuelva error', () => {
        const color = game.transformColorToNumber('aMarillo')
        expect(color).toBe('Error');
    });
    it('Comprobar en caso que le llegue otro color devuelva error', () => {
        const color = game.transformColorToNumber('amarillo')
        expect(color).toBe('Error');
    });
})
```

Funcionalitat:

-Game-resetValues

-Resetejar el valors del php de la pantalla, tant el Score a 0, com el nivell a 1, com el nom del jugador a buit “ “.

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

-Caixa Blanca /Test unitari

-Es Comprova que una vegada fet el resetValues, si els 3 valors de la pantalla se han resetejat per poder tornar a jugar.

Score = 0, Level = 1, Name = “ “

```
describe('Game-resetValues', () => {
    it('Comprobar que en caso de que se Gane/pierda la partida se reinicie los valores Score=0 , Nivel=1, name = " " ', () => {
        const score_nivel_name = game.resetValues();
        expect(score_nivel_name[0]).toBe(0);
        expect(score_nivel_name[1]).toBe(1);
        expect(score_nivel_name[2]).toBe(" ");
    });
})
```

Funcionalitat:

-GameProve1

-Faig la primera possibilitat que hi ha en el joc, fer tots el clicks be i “Guanyar”

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

- Caixa Blanca / Statement coverage

- Així queda el camí: (Els passos “ “ es simulen, ja que son visuals)

.Inicialitzem

. “Cliquem al boto Start”

.es genera el array

.“Introduïm Nom”

.Es transforma els números del array a colors

.“S’il·luminen”.

.Elecció Correcta (puja nivell), comprova si s’ha de seguir o ha guanyat (ha de seguir)

.“S’il·luminen”

.Elecció Correcta

.Elecció Correcta (son 2 perquè ja estem al nivell 2)

.Elecció Correcta (puja nivell), comprova si s’ha de seguir o ha guanyat (ha guanyat)

“ja que per tal de no repetit lo mateix tantes vegades al test he fet que nomes tingui 2 nivells”

."Missatge Has Guanyat"

.resetejaValues

.initialize

```
Game-prove
  √ initialize_BotonStart
  √ Array (1 ms)
  √ Ilumination_TransformNumberToColor
  Eleccion-Correcta-Nivel_1
    √ Eleccion Correcta en el nivel 1, subimos de nivel y puntuacion (1 ms)
    √ Eleccion Correcta en el nivel 1, subimos a nivel 2 y por lo tanto ya tenemos array 0 y 1
    Eleccion-Correcta-Nivel_2
      √ Eleccion Correcta en el nivel 2.1
      √ Eleccion Correcta en el nivel 2.2
      √ Comprobar que en caso de que se Gane/pierda la partida se reinicie los valores Score=0 , Nivel=1, name = " "
```

Test:

-Arxiu: IndiceJS.js

-Class: Game

- Caixa Blanca / Decision and Condition

-Aprofitant fent el test general, he comprovat els diversos camins possibles com per exemple que sigui correcte, però que també comprovi el camí incorrecte 1 a 1, i despres amb les condicions que no sigui ni Blau, ni Vermell, ni Groc, ha de ser Verd.

```
it('Eleccion Correcta en el nivel 1, subimos de nivel y puntuacion', () => {
    var Esperado = 0
    var Igual = false

    const NumberVerde = game.transformColorToNumber('Verde')
    if(Esperado != NumberVerde){expect(Igual).toEqual(false);}
    else{Igual = true; expect(Igual).toEqual(true);}

    expect(score).toBe(0)
    expect(nivel).toBe(1)
    const Number = game.transformColorToNumber('Verde')
    expect(Number).toBe(0);
    //Como la opcion ha sido correcta sumamos +1 punto por cada click correcto que se hace y tambien el nivel
    score+=1
    nivel+=1
    expect(score).toBe(1)
    expect(nivel).toBe(2)

});
it('Eleccion que habria sido incorrecta en el nivel 1, marcar "Rojo,Amarillo o Azul"', () => {
    var Esperado2 = 0
    var Igual2 = false

    const NumberRojo = game.transformColorToNumber('Rojo')
    if(Esperado2 != NumberRojo){expect(Igual2).toEqual(false);}

    const NumberAzul = game.transformColorToNumber('Azul')
    if(Esperado2 != NumberAzul){expect(Igual2).toEqual(false);}

    const NumberAmarillo = game.transformColorToNumber('Amarillo')
    if(Esperado2 != NumberAmarillo){expect(Igual2).toEqual(false);}

    if(NumberAmarillo != 0 && NumberAzul != 0 && NumberAmarillo !=0){expect(Igual2).toEqual(false);}
});
```

Funcionalitat:

-GameProve 2

- En aquest cas el primer nivell ja fallarà

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

- Caixa Blanca / Statement coverage

-Aixi queda el camí:

-Arxiu: IndiceJS.js

-Class: Game

- Caixa Blanca / Statement coverage

- Així queda el camí: (Els passos " " es simulen, ja que son visuals)

.Inicialitzem

. "Cliquem al boto Start"

.es genera el array

."Introduïm Nom"

.Es transforma els números del array a colors

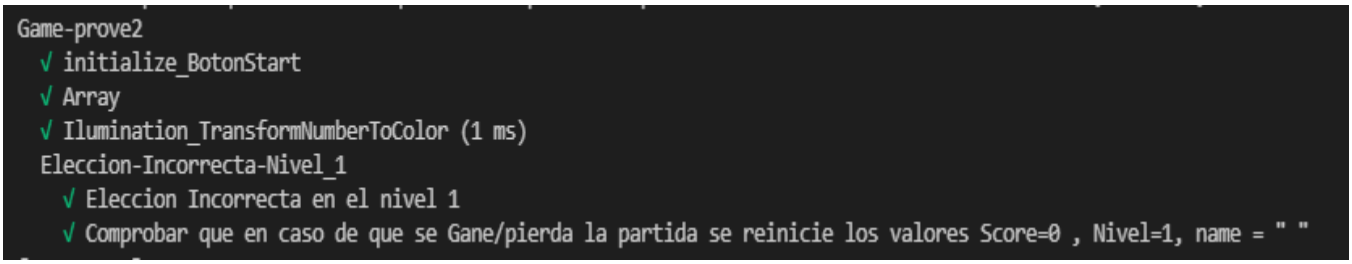
."S'il·luminen".

.Elecció Incorrecta

."Missatge Has Perdut"

.resetejaValues

.initialize



Funcionalitat:

-GameProve 3

- En aquest cas el primer nivell ho farà correctament però el 2n nivell no

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

- Caixa Blanca / Statement coverage

- Així queda el camí:

.Inicialitzem

. “Cliquem al boto Start”

.es genera el array

.“Introduïm Nom”

.Es transforma els números del array a colors

.“S’il·luminen”.

.Elecció Correcta (puja nivell), comprova si s’ha de seguir o ha guanyat (ha de seguir)

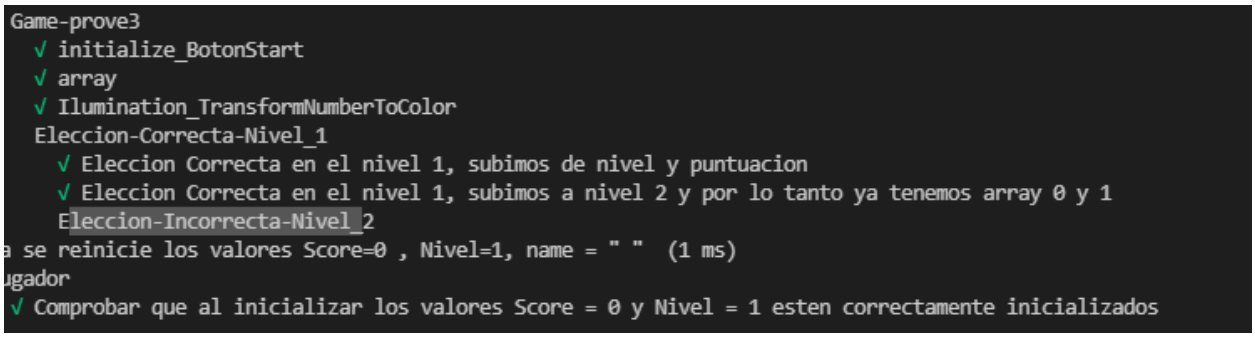
.“S’il·luminen”

.Elecció Incorrecta

.”Missatge Has Perdut”

.resetejaValues

.initialize



Funcionalitat:

-GameProve 1-2-3

- En aquest cas el primer nivell ho farà correctament però el 2n nivell no

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Game

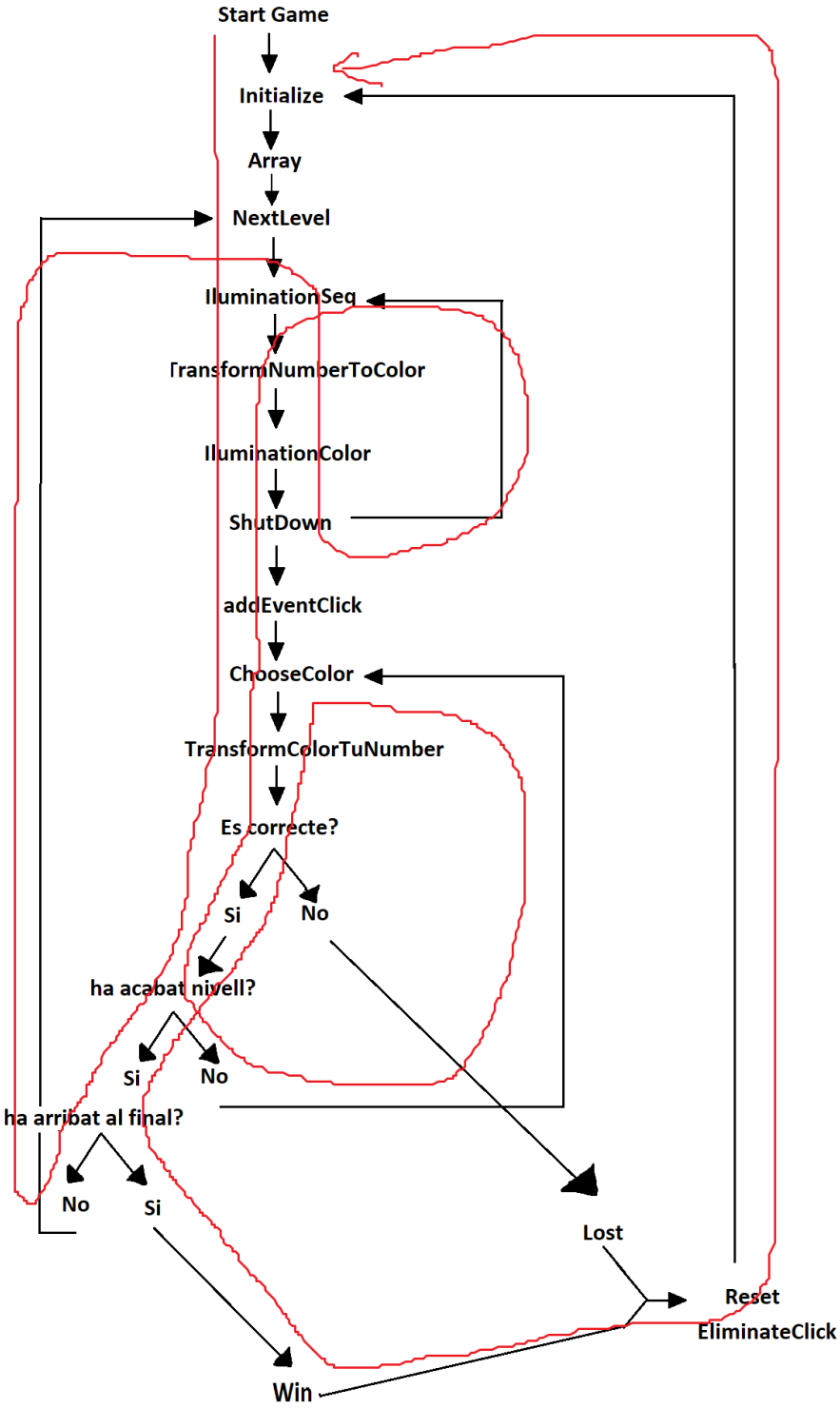
-Caixa Blanca / Path Coverage / loop Testing

Amb els GameProve 1-2-3 ja he comprovat tots el casos que es poden donar en una petita escala, es a dir, jo ho he comprovat amb un array de 2, es a dir, hi ha 2 nivells en el joc.

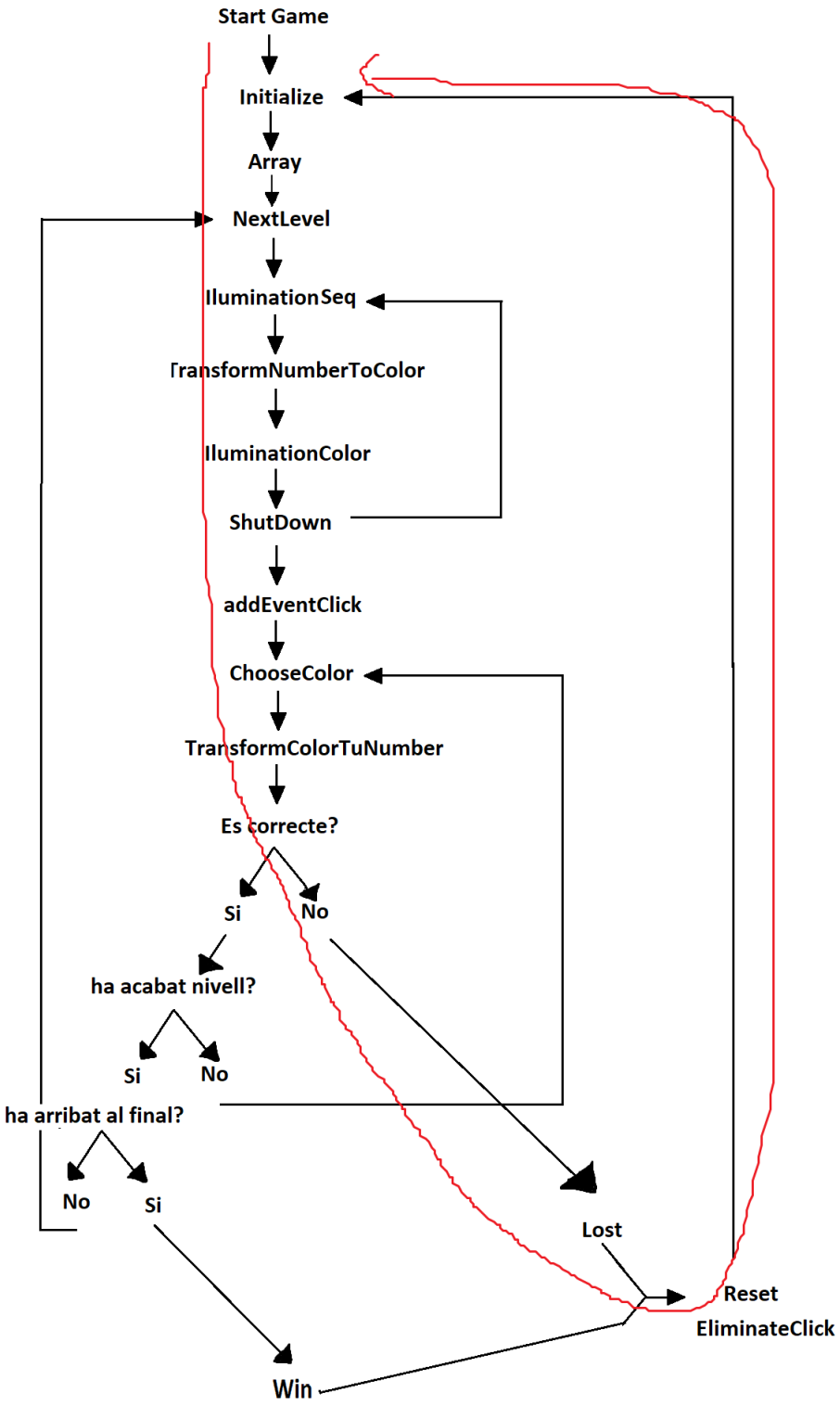
I també amb els prove he provat el loop testing, tant evitant el loop com fent una passada per cadascun

-per tant les meves possibilitats són:

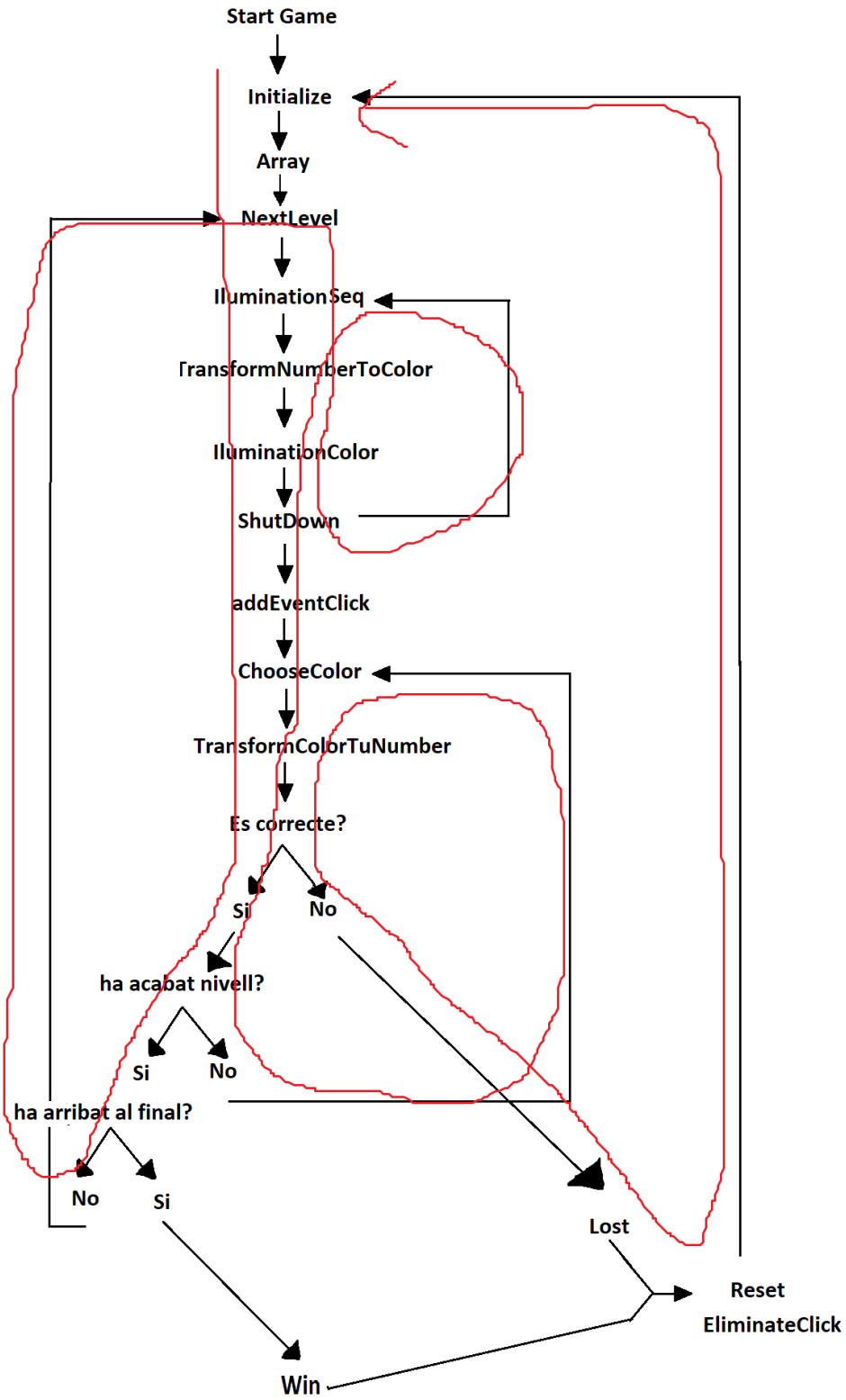
Game_Prove_1


Start Game
Initialize
Array
NextLevel
IlluminationSeq
TransformNumberToColor
IlluminationColor
ShutDown
addEventClick
ChooseColor
TransformColorTuNumber
Es correcte?
Si
No
ha acabat nivell?
Si
No
ha arribat al final?
No
Si
Lost
Reset
EliminateClick
Win

Game_Prove_2


Start Game
Initialize
Array
NextLevel
IluminationSeq
TransformNumberToColor
IluminationColor
ShutDown
addEventClick
ChooseColor
TransformColorTuNumber
Es correcte?
Si
No
ha acabat nivell?
Si
No
ha arribat al final?
No
Si
Lost
Reset
EliminateClick
Win

Game_prove_3


Start Game
Initialize
Array
NextLevel
IluminationSeq
TransformNumberToColor
IluminationColor
ShutDown
addEventClick
ChooseColor
TransformColorTuNumber
Es correcte?
Si
No
ha acabat nivell?
Si
No
ha arribat al final?
No
Si
Lost
Reset
EliminateClick
Win

Funcionalitat:

-Jugador

-Al inici de la partida creem una classe jugador amb dades simples, score i name, i amb aquesta classe fem que escrigui el nom que es veurà per pantalla, i que comprovi que s'escriu alguna cosa si o si.

Localització: /SimonDice/Main/View/JS/IndiceJS.js

Test:

-Arxiu: IndiceJS.js

-Class: Jugador

- Caixa Blanca / test unitari

- En el test es comprova que tant el nom com el score han d'estar inicialitzats abans de donar la possibilitat d'escriure el nom per pantalla.

```
describe('Jugador', () => {
    var jugador = new Jugador();

    //constructor()
    it('Comprobar que al inicializar los valores Score = 0 y Nivel = 1 esten correctamente inicializados', () => {
        const name = jugador.getName();
        expect(name).toBe("");
        const score = jugador.getScore();
        expect(score).toBe(0);
    });

});
```

Un diagrama explicatiu del passos que segueix el joc