

# SameGame E2 - Diseño

Santiago Romero, Diego Bulla, David Lopez, Ricardo Bernal  
Unión del grupo Pandemia y Backslash

October 2020

## 1 Introducción

El presente documento está orientado al diseño y análisis correspondiente a la entidad principal "Jugador" en nuestra versión del juego. En primer instancia, se describe la complejidad temporal y espacial de la lógica (véase Pseudocódigo) del método principal del jugador definida como "jugar", el cual permite llevar a cabo un juego según un tablero dado y retornara las jugadas realizadas para terminar el juego.

En segundo lugar, se describe la lógica del código, específicamente el funcionamiento del mapa dentro de la metodología para resolver un tablero.

## 2 Análisis

jugar()

- $T.E = n \times m$ 
  - $n$  = Cantidad de filas del tablero.
  - $m$  = Cantidad de columnas del tablero.
- $t(n) = O((n^2)^k)$ .
  - $K$  = cantidad de subconjuntos de casillas del mismo color formadas en una partida.
- $S(n) = O(C)$ .
  - $C$  = Cantidad de colores.

## 3 Pseudocodigo

---

**Algorithm 1** jugar

---

```
1: procedure JUGAR(tablero) ▷ El objeto Tablero dado por el robot
2:   seguirJugando  $\leftarrow$  True
3:   valorJugada  $\leftarrow$  0
4:   list  $\leftarrow$  Casilla  $\leftarrow$  listaJugadas  $\leftarrow$  0
5:   while seguirJugando do ▷ seguirJugando = True
6:     realizaJugada  $\leftarrow$  False
7:     filas  $\leftarrow$  tablero.getFilas()
8:     for i  $\leftarrow$  0 to filas do
9:       columnas  $\leftarrow$  tablero.getColumns()
10:      for j  $\leftarrow$  columnas to 0 do
11:        color  $\leftarrow$  tablero.colorCasilla(i, j) ▷ color es entero (int)
12:        h  $\leftarrow$  hash(color)
13:        if H.contieneLlave(h)  $\neq$  True then
14:          casilla  $\leftarrow$  Casilla(i, j)
15:          H.insert(h, casilla)
16:          if color == H.values().toArray()[0] then
17:            if tablero.tieneVecinos(i, j) then
18:              nCasilla  $\leftarrow$  Casilla(i, j)
19:              puntaje  $\leftarrow$  tablero.efectuarJugada(nCasilla)
20:              listaJugadas.add(nCasilla)
21:              realizaJugada  $\leftarrow$  True
22:              break
23:          if realizaJugada then
24:            break
25:      for llave to len(H) do
26:        A  $\leftarrow$  H.get(llave)
27:        if A.getX  $\neq$  -1 & A.getY  $\neq$  -1 then
28:          if realizaJugada = False then
29:            nCasilla  $\leftarrow$  Casilla(A.x, A.y)
30:            puntaje  $\leftarrow$  tablero.efectuarJugada(nCasilla)
31:            listaJugadas.add(nCasilla)
32:            realizaJugada  $\leftarrow$  True
33:        B  $\leftarrow$  A
34:        B.setX(-1)
35:        B.setY(-1)
36:        H.put(llave, B);
37:  return listaJugadas
```

---

## 4 Explicación del mapa

Para comenzar debemos tener en cuenta que este mapa se encuentra completamente vacío y se genera dependiendo del número de colores que existen en el tablero. Para la explicación de este se tomará en cuenta el siguiente juego con el cual se ira explicando paso a paso como se desarrolla.

A continuación, tenemos un ejemplo de un tablero en el cual tenemos 4 colores.

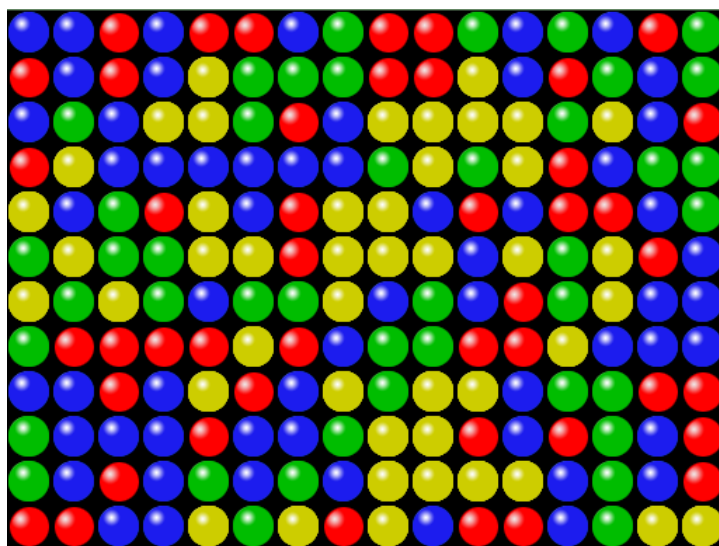


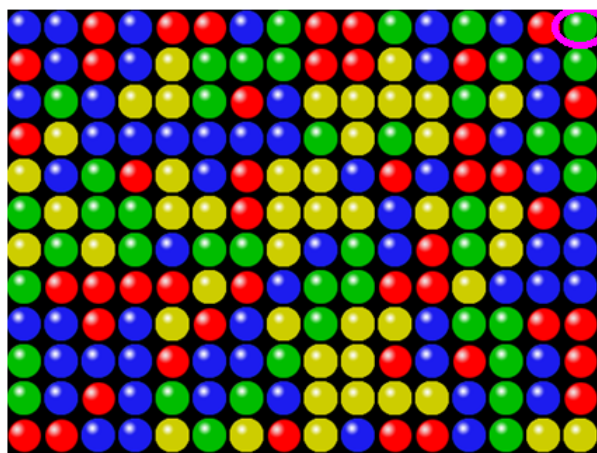
Figure 1: Descripción

Basados en estos 4 colores llenaremos en primer lugar el mapa de la siguiente manera, donde hay que tener en cuenta que los colores todavía en este momento no han adquirido ninguna prioridad lo cual es de gran importancia para nosotros. Es decir, solo se sabe que estarán los cuatro colores en los recuadros seleccionados con el recuadro naranja.

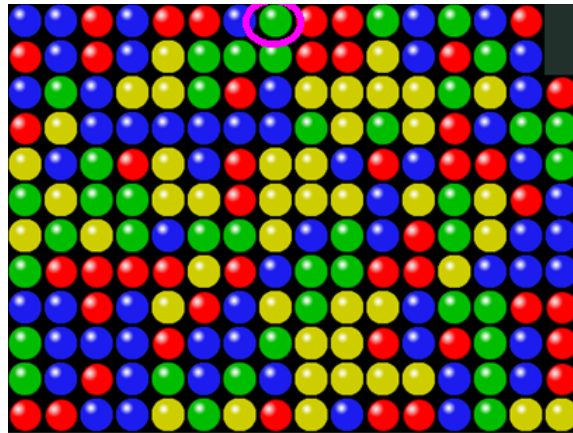
Color	Fila	Columna

Ahora para el llenado de este mapa que tenemos acá se realiza la siguiente verificación la primera casilla (arriba a la derecha subrayada de color rosado) se verifica si esta se puede jugar. De ser correcto este color (en este caso el verde) se selecciona como color con la mayor prioridad. En efecto luego de haber realizado esta periodización se remite a ejecuta esta jugada, eliminando para este caso las dos verdes que están allí.

Supongamos por un momento que en esa primera esquina no existiera ninguna jugada este color se selecciona como color secundario y así sucesivamente hasta encontrar un color con una jugada.



Luego de haber realizado esta jugada se comienza de nuevo desde el mismo punto de partida (la esquina superior derecha) de allí comenzamos a revisar nuestras posibilidades para jugar con el color verde ya que es nuestra prioridad. La evaluación sigue hacia la izquierda donde encontramos que hay un color rojo, el cual se convierte automáticamente como color secundario en nuestro orden de prioridades, luego está el color azul del mismo modo entrando al mapa como color terciario en el orden de prioridad (guardando siempre la posición de este). Cabe resaltar que esta posición inicial solo se guarda para la primera recursión que hagamos sobre el color ya que más adelante estaremos guardando la posición de una posible jugada.



Para mejor entendimiento de este proceso, veamos cómo queda el mapa en este momento.

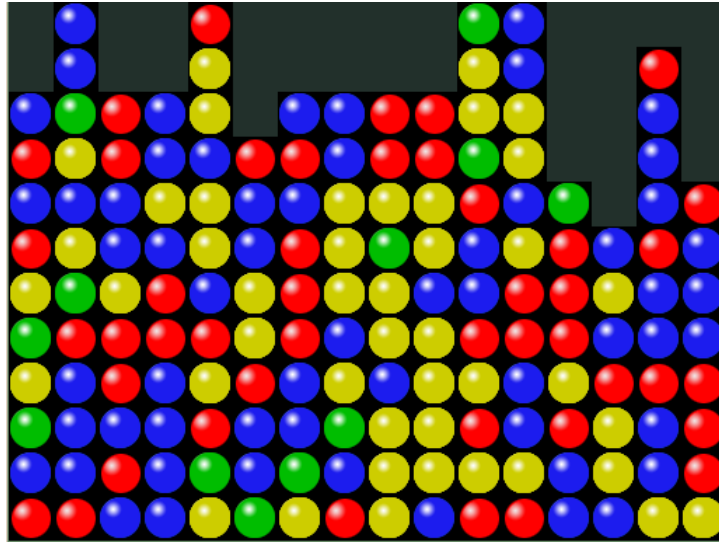
- Antes de llegar a la jugada del siguiente verde:

Color	Fila	Columna
Verde		
Rojo	0	14
Azul	0	13

- Después de la jugada del verde:

Color	Fila	Columna
Verde	0	7
Rojo	0	14
Azul	0	13

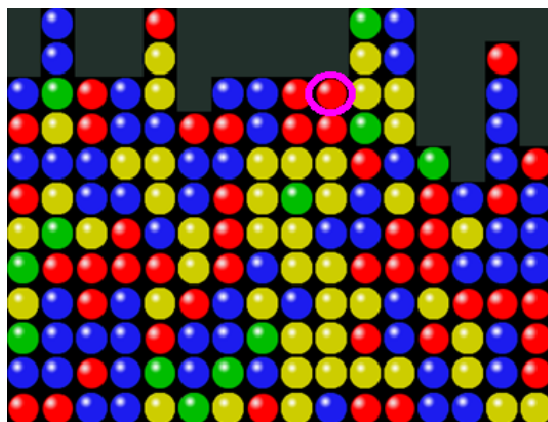
Es importante resaltar que el amarillo todavía no ha tenido posición ya que si vemos bien el juego el algoritmo no ha llegado a ningún amarillo todavía. Así se ira ejecutando el programa hasta que llegue el momento que no encuentre más jugadas del color prioritario (verde). Veamos como seria este tablero. También es importante saber que la posición de los colores que no es el prioritario en ese momento es para que en caso de que no exista ninguna jugada (como se verá a continuación) con el color prioritario se realice una jugada con el color secundario.



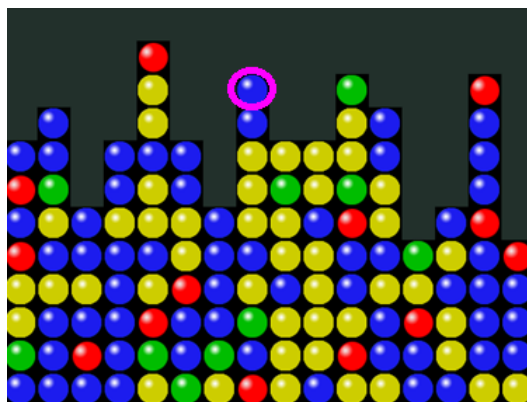
Como se ve, ya no tenemos en este momento posibilidades de jugar el color verde. Por lo tanto, nos remitimos a como quedo nuestro mapa para ver cuál es nuestro segundo color en prioridad y cuál es su posición para así eliminarlo.

Color	Fila	Columna
Verde		
Rojo	2	9
Azul	0	11
Amarillo	1	10

Como se puede ver el verde no tiene valores ya que no existe ningún movimiento disponible en este momento. Ahora el paso que sigue es que se va a eliminar la roja que se encuentra en la fila y columna mencionadas en el recuadro de arriba.



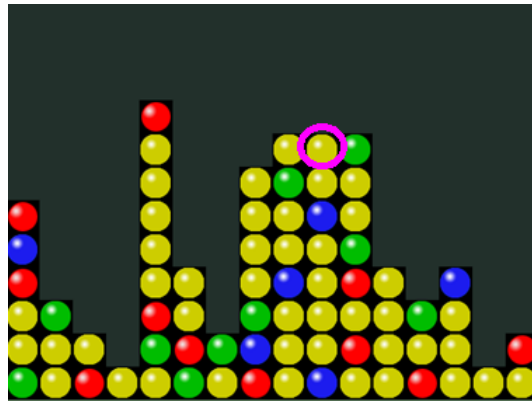
Ahora luego de haber eliminado todas las rojas y verdes posibles se tiene el siguiente tablero.



Para esta oportunidad el mapa quedaría de la siguiente manera:

Color	Fila	Columna
Verde		
Rojo		
Azul	2	7
Amarillo	2	4

Como se puede apreciar se tiene seleccionada la azul que esa sería la siguiente jugado ya que no existen más verdes o rojas. Ahora que hemos terminado con las azules seguiremos con las amarillas.



Mapa:

Color	Fila	Columna
Verde		
Rojo		
Azul		
Amarillo	4	9

Paso siguiente como se puede ver es eliminar esa sección de amarillas. Sucesivamente hasta terminar que el tablero quede terminado o hasta que no existan más movimientos disponibles.