SUDOKU

Plan :

- Project presentation
- Concept of the game
- Tasks to do
- Class creation
- Interface design and creation

Project Description

- The main goal of this project it to make in C++ programming language a program that generates and completes Sudoku grills of different sizes such as 4x4, 9x9, 12x12 and also 16x16, All that using mudular programming tools such as (Classes, inheritance, method independency) and also making an user friendly graphic interface.

Concept of The SUDOKUGAME:

- What is Sudoku?
- A Sudoku puzzle is defined as a logic-based, number-placement puzzle. The objective is to fill a 9×9 grid with digits in such a way that each column, each row, and each of the nine 3×3 grids that make up the larger 9×9 grid contains all of the digits from 1 to 9. Each Sudoku puzzle begins with some cells filled in. The player uses these seed numbers as a launching point toward finding the unique solution.
- The Rules of Sudoku While solving Sudoku puzzles can be significant challenge, the rules for traditional solution finding are quite straight forward:
- 1-Each row, column, and nonet can contain each number (typically 1 to 9) exactly once.
- 2- The sum of all numbers in any nonet, row, or column must match the small number printed in its corner.

Goals

- Creating classes that'll do :
  - Sudoku grills generation.
  - Errors detection.
  - Input/ output management.
  - Indications.
  - Reset the grills.
- Creating a graphic interface that'll insure:
  - Level Choice.
  - Fill and delete case content.
  - Time chronometre .
  - Saving games.

Class creation.

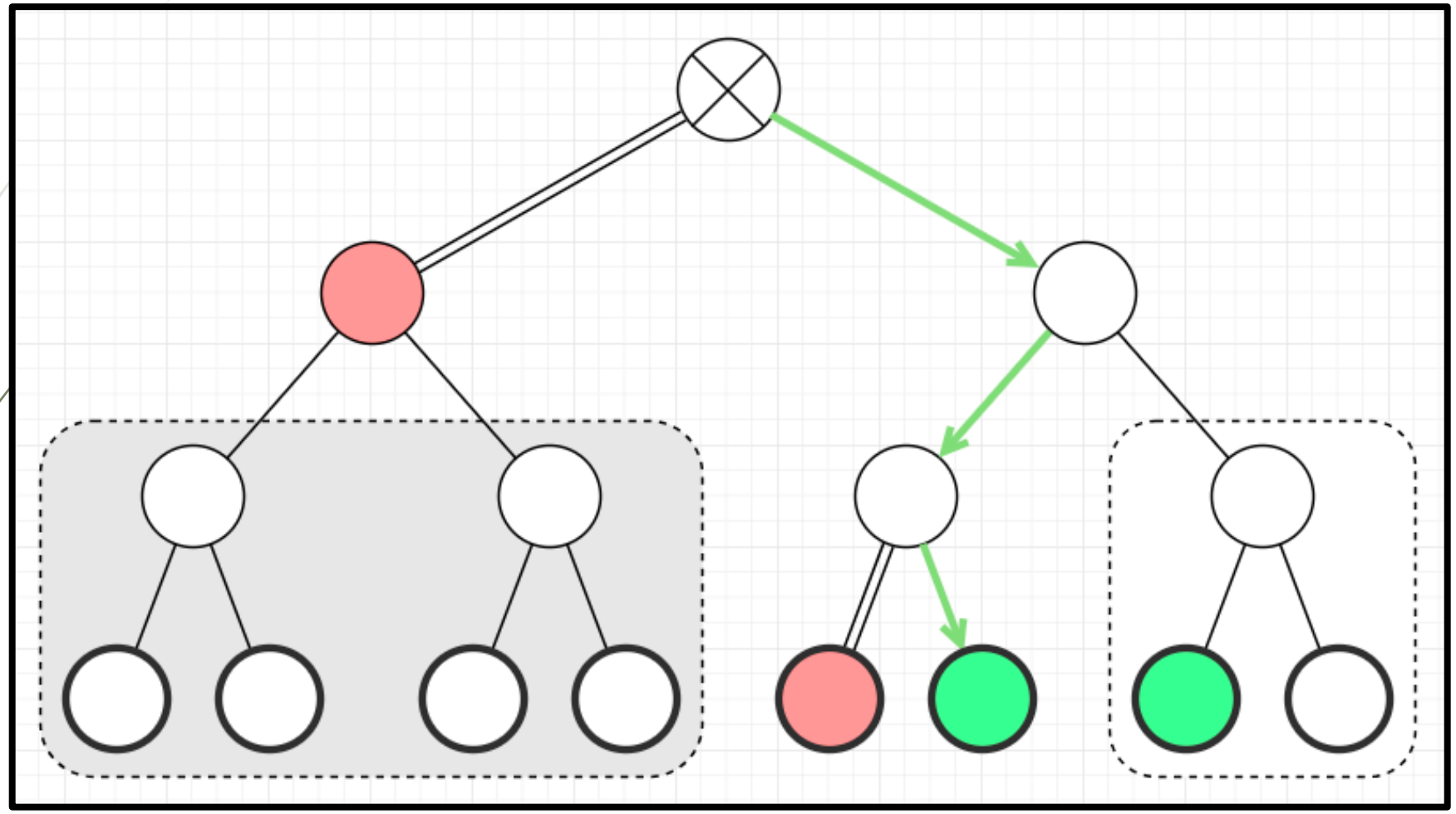
This SUDOKU game consists of six levels which require 4 classes:

- **Class KIDS.**
- **Class SUDOKU 9 x 9.**
- **Class SUDOKU 12 x 12.**
- **Class SUDOKU 16 x 16.**

Class KIDS:

- This class is responsable for managing all the elements of the kids level including the generation of the grill and the solving :
  - **-kids() : Constructor**
  - **-element_write()**
  - **-element_read()**
  - **-element_read_solution()**
  - **-clear_grille()**
  - **-reset_grille()**
  - **-goodcolumn()**
  - **-goodrow()**
  - **-goodblock()**
  - **-recherche()**
  - **-resultat()**
  - **-~kids() destructor**

BACKTRACKING



BACKTRACKING FOR SUDOKU9 x 9 **?**

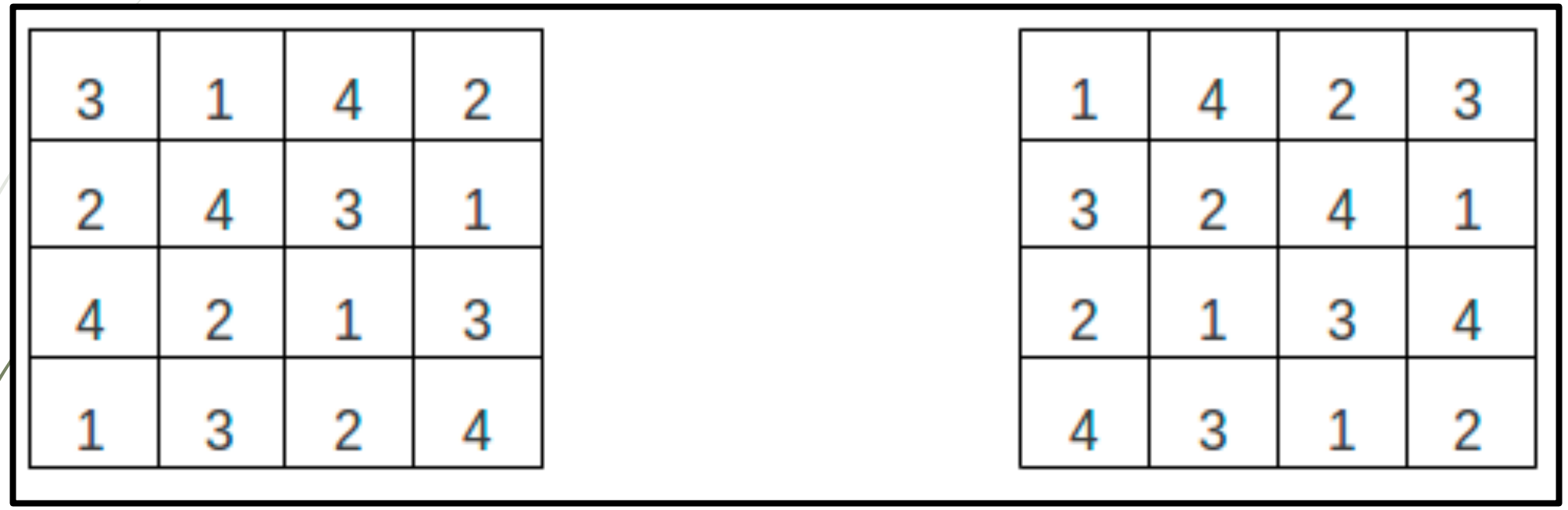
Problem!!

5	3	1	2	7	6	8	9	4
6	2	4	1	9	5	2		
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Class SUDOKU 9 x 9 :

- This class contains the method responsible for the 9x9 SUDOKU level.
  - **-Sudoku() : Constructor**
  - **-element_write()**
  - **-element_write_backup()**
  - **-element_write_solution()**
  - **-element_read()**
  - **-element_read_backup()**
  - **-element_read_solution1()**
  - **-clear_grille()**
  - **-reset_grille()**
  - **-goodcolumn()**
  - **-goodrow()**
  - **-goodblock()**
  - **-~sudoku() destructor**

Permutations



3	1	4	2
2	4	3	1
4	2	1	3
1	3	2	4

1	4	2	3
3	2	4	1
2	1	3	4
4	3	1	2

Example : 90deg permutation

Class SUDOKU 12 x 12:

- This class contains the method responsible for the MASTER SUDOKU level, it contains :
  - **Sudoku() : Constructor**
  - **element_write()**
  - **element_write_backup()**
  - **element_write_solution()**
  - **element_read()**
  - **element_read_backup()**
  - **element_read_solution1()**
  - **clear_grille()**
  - **reset_grille()**
  - **goodcolumn()**
  - **goodrow()**
  - **goodblock()**
  - **~sudoku() destructor**

Classe SUDOKU 16 x 16:

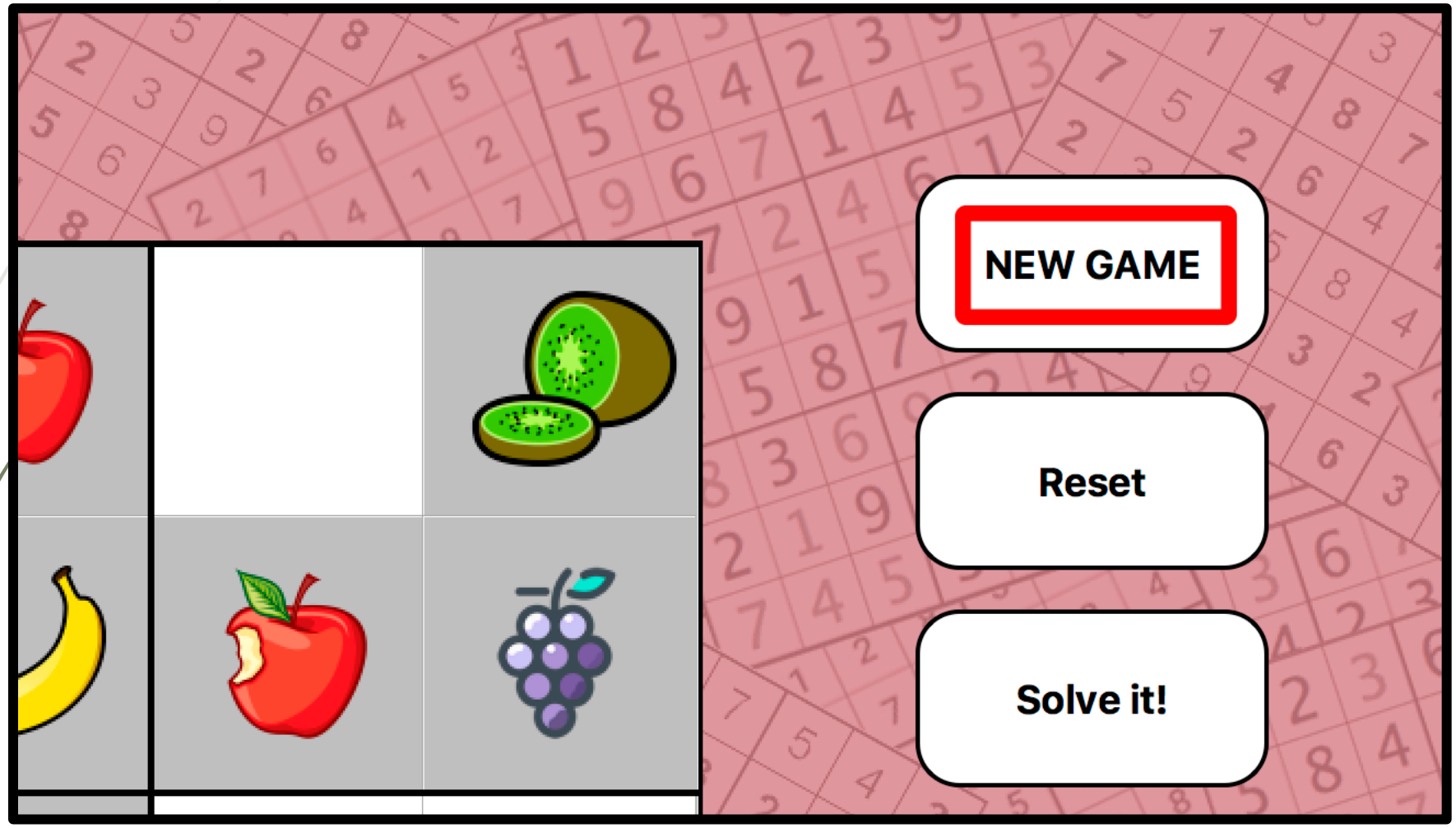
- This class contains the method responsible for the KING SUDOKU level, it contains :
  - **-Sudoku() : Constructor**
  - **-element_write()**
  - **-element_write_backup()**
  - **-element_write_solution()**
  - **-element_read()**
  - **-element_read_backup()**
  - **-element_read_solution1()**
  - **-clear_grille()**
  - **-reset_grille()**
  - **-goodcolumn()**
  - **-goodrow()**
  - **-goodblock()**
  - **-~sudoku() destructor**

Design & interface creation

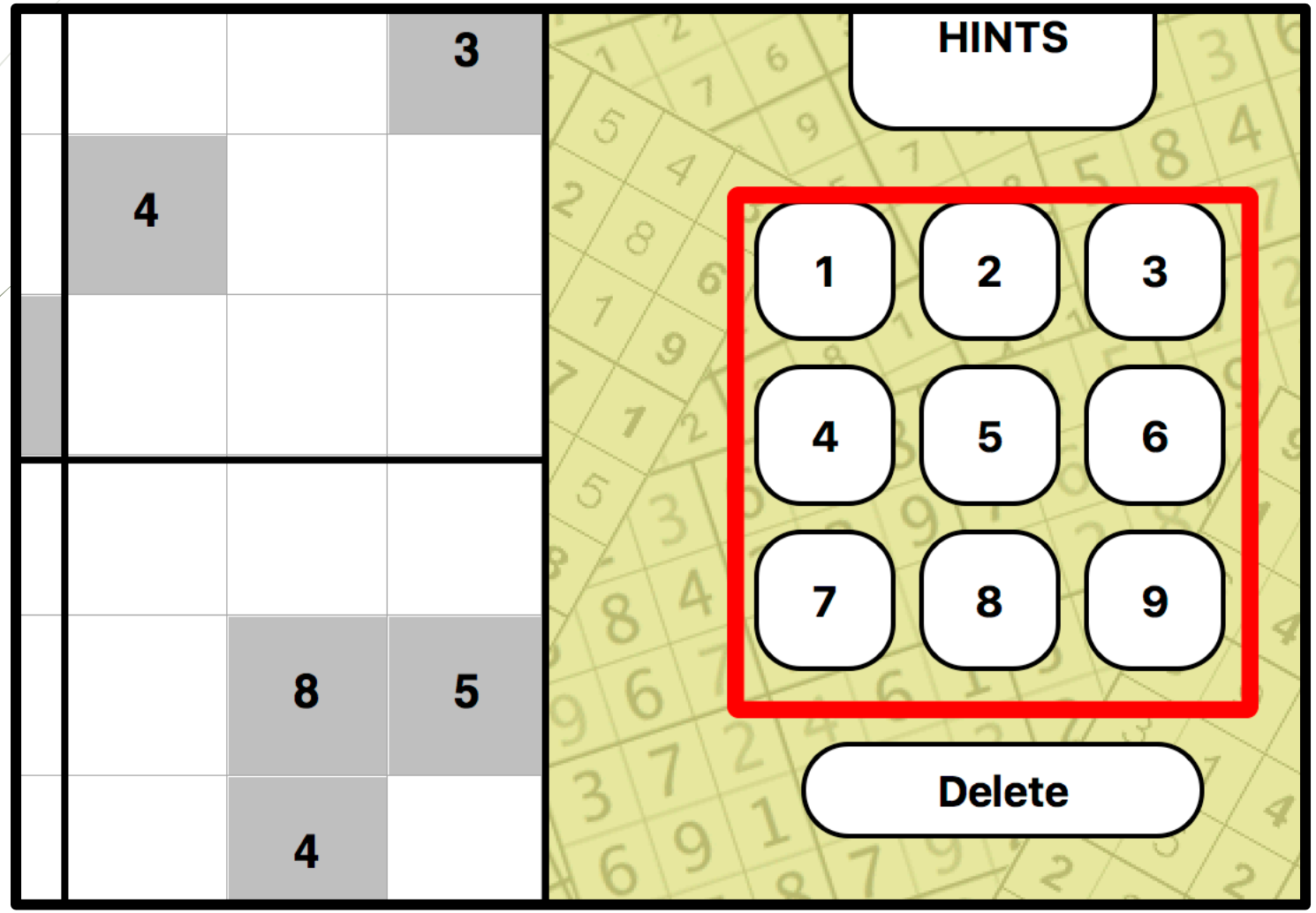
The interfaces used contains many methods (events) :

- **On_newgame_clicked()**
- **On_Number_clicked()**
- **On_Solveit_Clicked()**
- **UpdateTime()**
- **On_erase_clicked()**
- **On_menu_clicked()**
- **On_Indication_clicked()**
- **On_Save_clicked()**
- **On_Reload_clicked()**

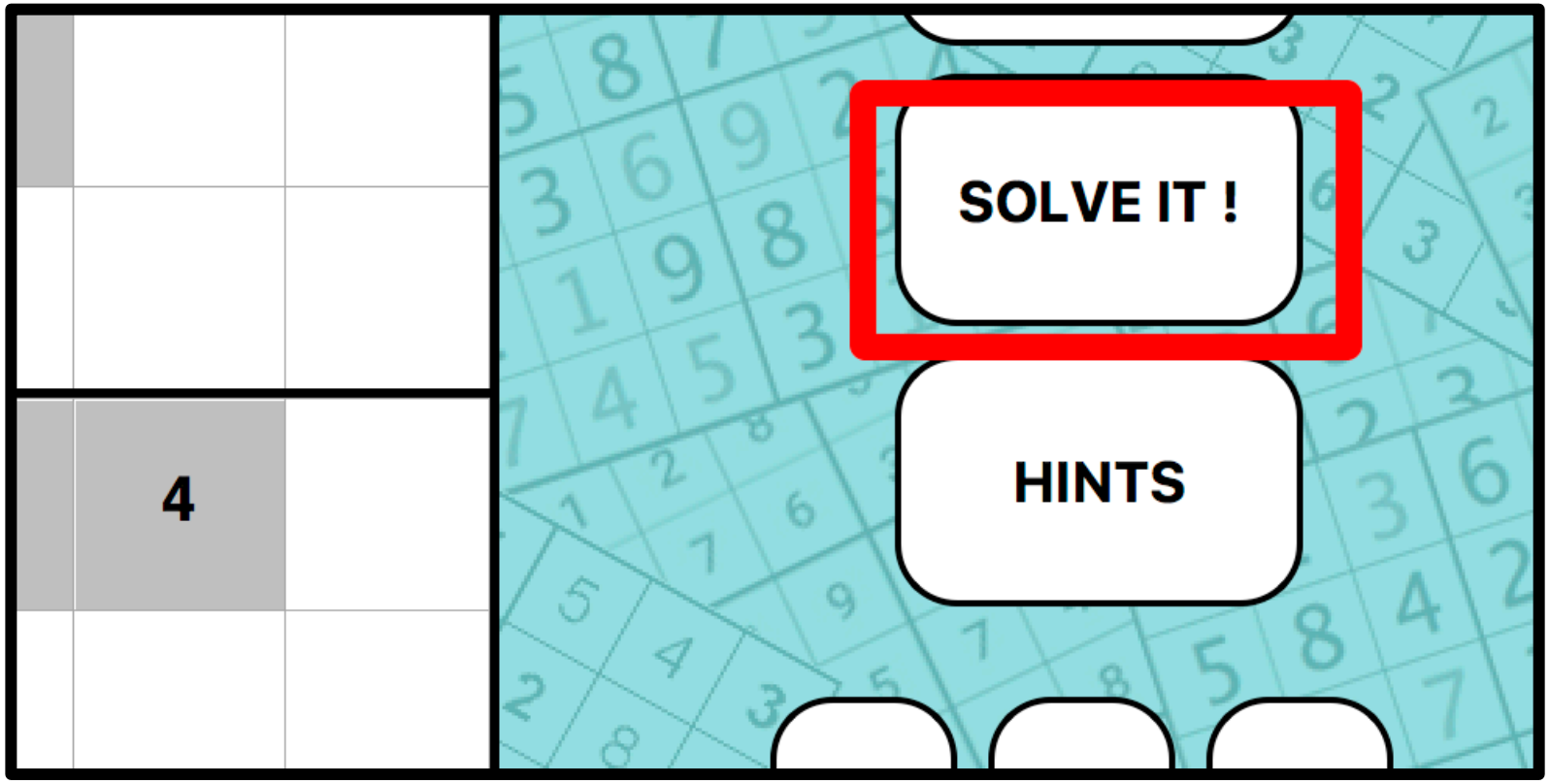
On_New_Game()



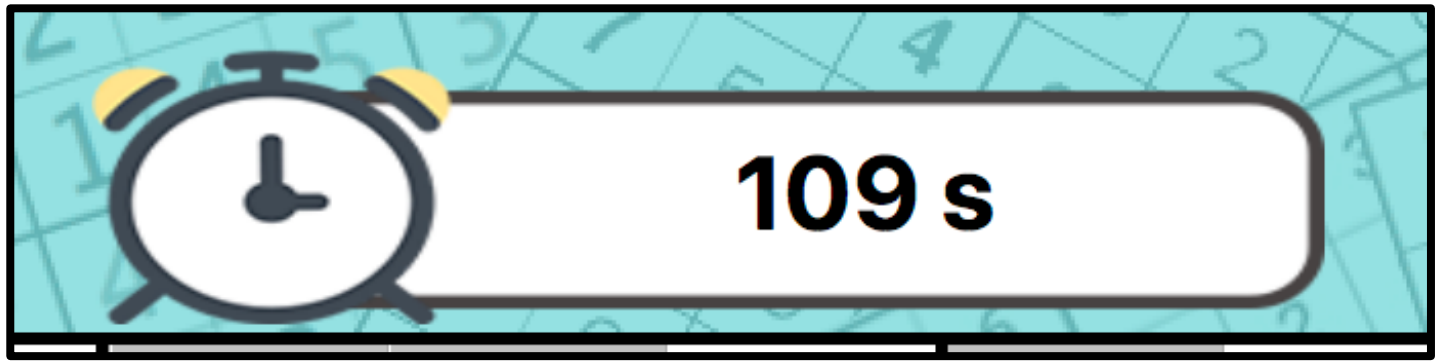
On_Number_clicked()



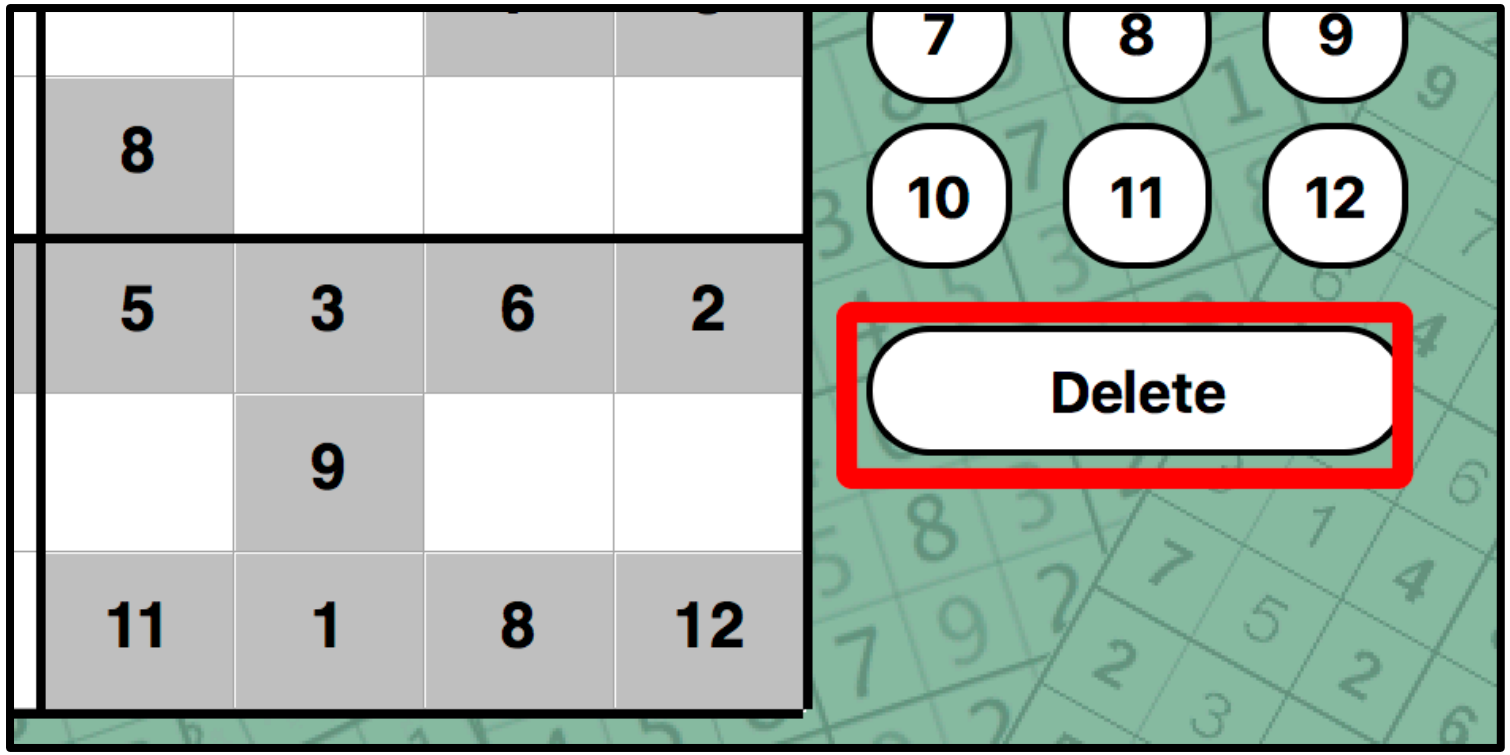
On_Solveit_clicked()



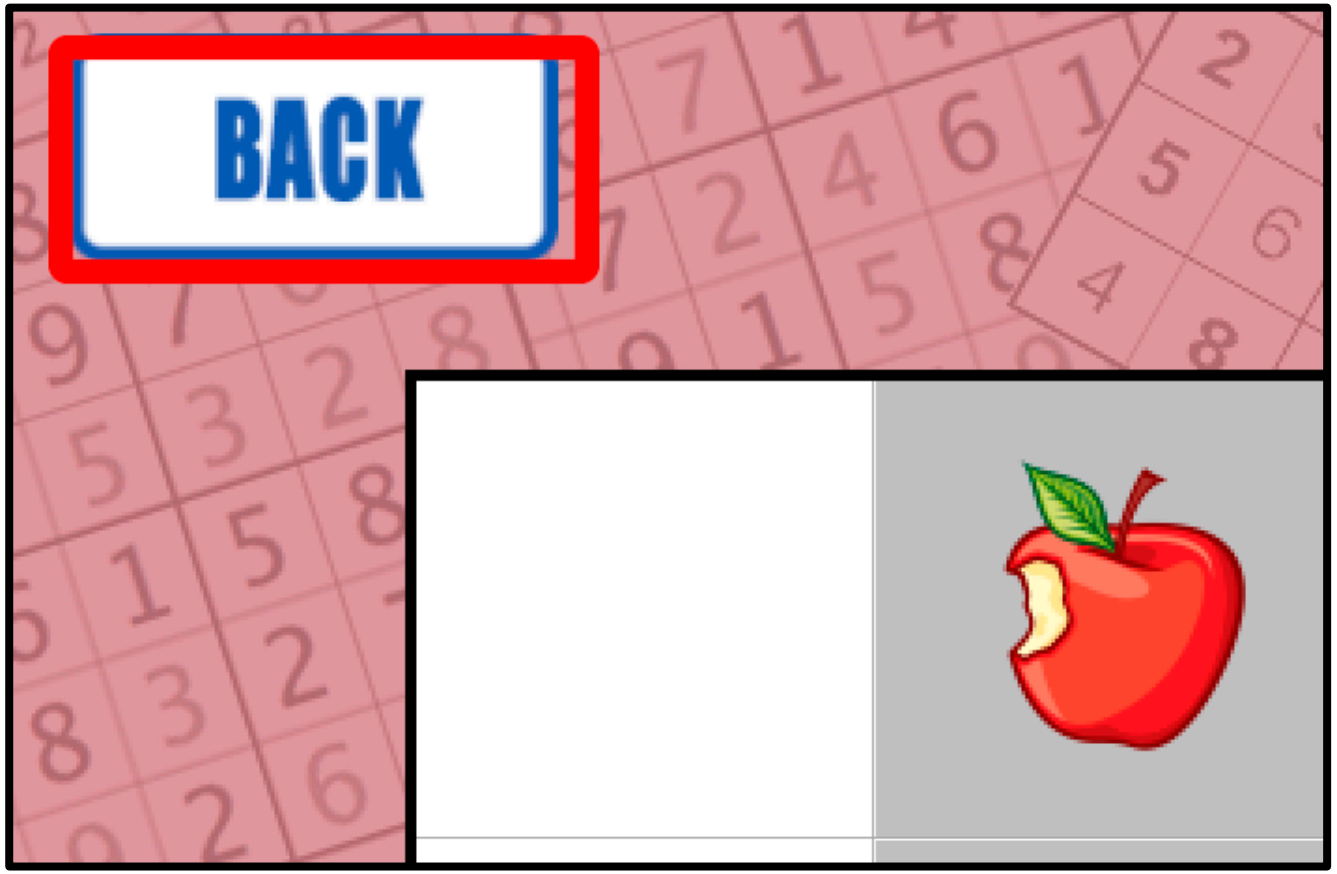
UpdateTime()



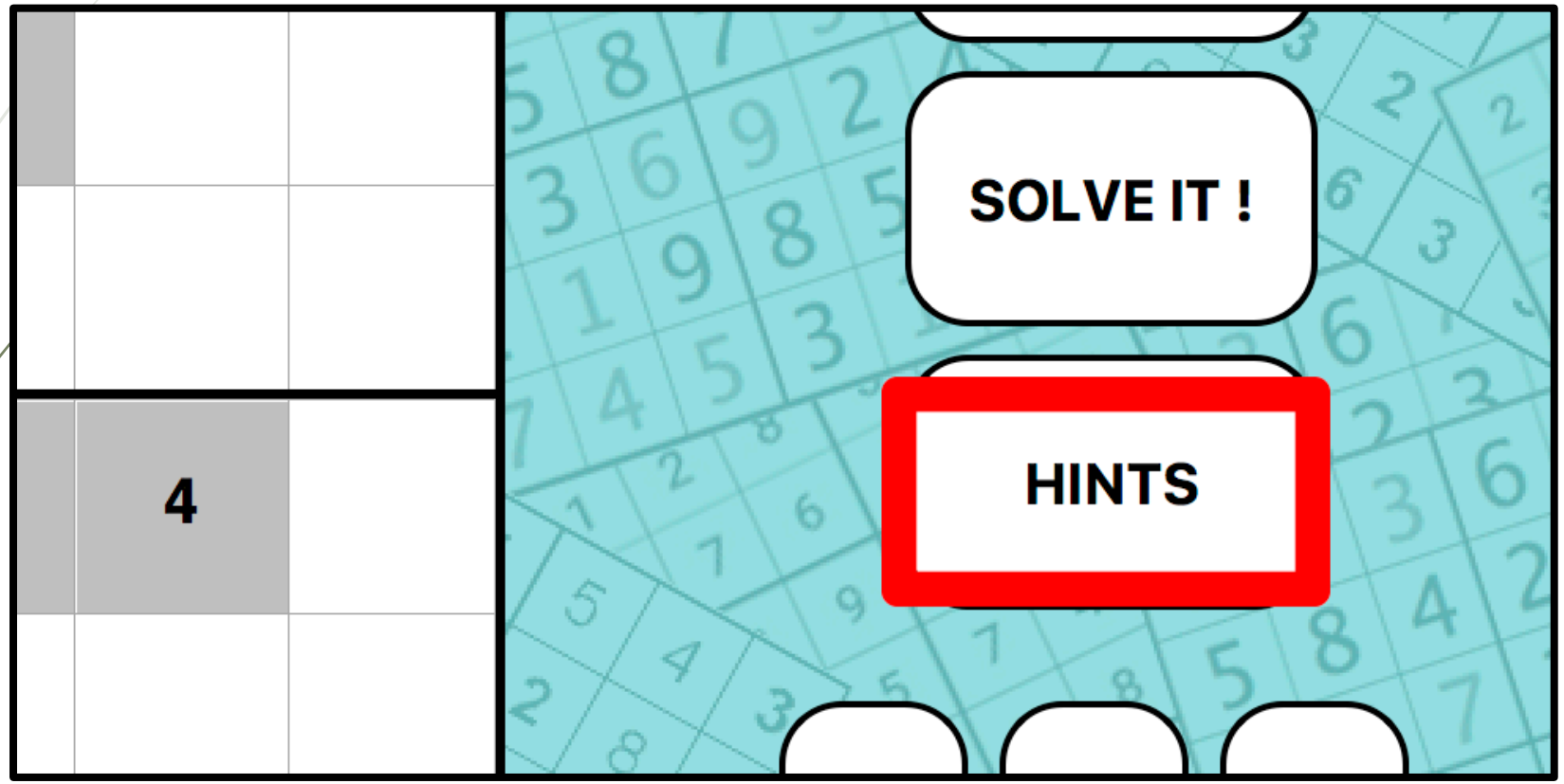
On_Erase_clicked()



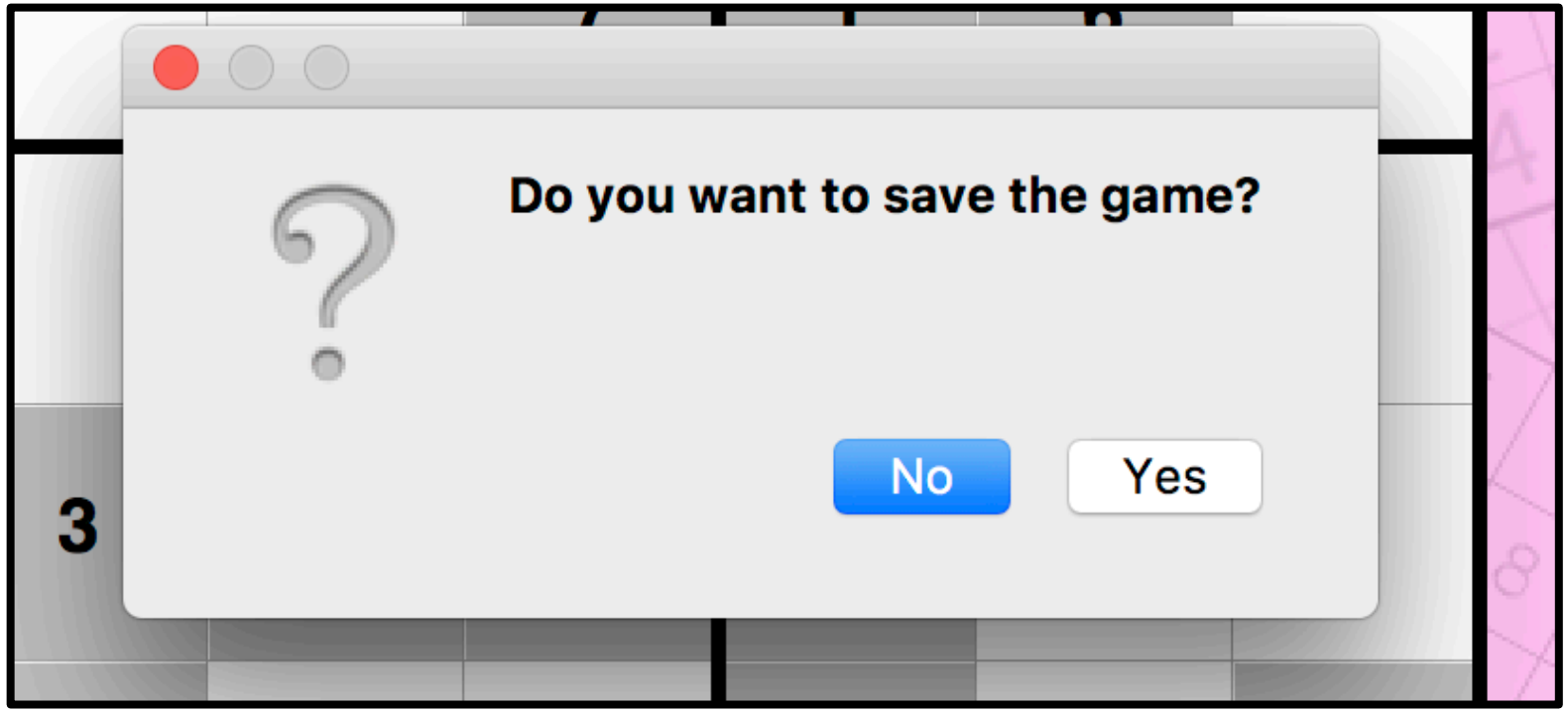
On_Menu_clicked()



On_Indication_clicked()



On_Save_clicked()



On_Reload_clicked()