

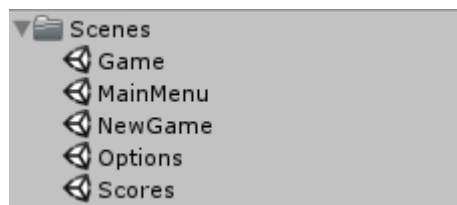
Sudoku v1.1

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

Sudoku is a old logic game with numbers created in Japan over 1000 years ago. Goal of the game is to fill 9x9 square box with numbers. There are 9 3x3 square boxes in a large box, and each 3x3 square box can hold numbers from 1 to 9 (1,2,3,4,5,6,7,8,9) without repeating one number twice. That means that all 9 boxes must contain all numbers from 1 to 9, and the whole 9x9 box must contain each number from 1 to 9 nine times (9x1, 9x2, 9x3, 9x4, 9x5, 9x6, 9x7 ,9x8, 9x9). The 9x9 box also has 9 rows and 9 columns and each row and each column can hold numbers from 1 to 9 without repeating. When all squares are filled the puzzle is solved and the play time is recorded.

There are three difficulty levels - easy, medium and hard. Each level best score is saved. Each new game is unique and different, puzzle solution is saved after puzzle generation so if you cant solve it you can see the solution by clicking on "Solve" button. If you want to clear are the numbers you entered in the box you can click on "Clear" button. Only the numbers you have entered will be erased but not the starting gray numbers. Clicking on a "New" button will generate new puzzle with the same selected difficulty level. Clicking on a "Quit" button will load main menu and if you click on "Quit" on main menu then the game will quit. Filling the squares is very easy, select empty field or a field with a number you have entered and select available number on the bottom of the screen. If you want to switch menu just click on the "X" mark next to number nine. To close the menu just select a field.

Whole game is created with 5 scenes just to make it as simple as possible. Each menu is a different scene, game scene is where game logic is.



When you load Game scene select Main Camera game object, you will see a script attached to the object called Game.



This script is where all game logic is. There are few fields you should not edit because it could corrupt the game.

- Box1 – Box9 – are used as a field containers and **SHOULD NOT BE EDITED**
- Num – is used as a texture array for field numbers
- Lock Num – is used as a texture array for locked field numbers, you cant replace textures in Num or Lock Num with your own number textures.
- Buttons – is a GUITexture array used for showing, enabling and disabling number buttons.
- Selected - selected field object
- Num Buttons – game object containing numbers in scene
- Game Buttons – game object containing game menu buttons in

the scene

- Solved – has the puzzle been solved
- Tex Solved – GUITexture for displaying when the puzzle is solved
- Tex Failed – GUITexture for displaying when the player quits the game
- Gen – GUITexture showing when generating puzzle
- Click Sound – simple click sound

In Options menu there is an option to turn off the music but there is no music playing object in the game, so if you want one you can add it to the Game scene. To check if music is on or off just get string from `PlayerPrefs.GetString("musicvolume" , " on");`. If "on" then set audio volume of the audio source to 1.0f, if off set it to 0.0f.

UPDATE v1.1

In this update I have removed puzzle generator due to long waiting time on slower machines and mobile devices. Now there is a simple prefab in the "Game" scene called "Database" . This prefab contains generated puzzles in one simple array, demo version has 100 puzzles but you can add any number of puzzles you need. To add new puzzles just go Sudoku > Prefabs folder and select Database prefab. Expand Puzzle array and change number from 100 to any number of puzzles you want to add, after you finish just save your scene and project.

Puzzle Database (Script)	
Script	PuzzleDatabase
▼ Puzzle	
Size	100
Element 0	7348592162687415939516327846179238
Element 1	9714328565821673494368957212683749
Element 2	8123976545368421797946518233512649
Element 3	7816359429467128355324987166945831
Element 4	4389625179214576385673184298597243
Element 5	2763148955348791628192653471429586
Element 6	4789123655364879129123568472638715
Element 7	5437198627986251341264835794893627
Element 8	8596432714132759862679815435761943
Element 9	8769452134523186791936278452385719
Element 10	6213589749471628538354792617936154
Element 11	9614872353452961788723154961347698
Element 12	6943817255312794867284653193827469
Element 13	8427195633156248976973584121648327
Element 14	3925741868651397244178629357862435
Element 15	8274965319361582475417326896759238
Element 16	9621837454152769838375942163917258
Element 17	3768421954591763821283596478472915
Element 18	7521869349814375264639258171296543
Element 19	6437259815176984328923147653549621
Element 20	2796481351567932848432159675243867
Element 21	6938752415149327868721645394816593
Element 22	5241639876387594211978425637493862
Element 23	9846137522157893647635421898721956
Element 24	7593164286412985378325479612837541
Element 25	1926853473641978255782439162457396
Element 26	1673458929428765318351296747192384
Element 27	2318956746492375188574162935236741
Element 28	9645831273576128492187946355721493
Element 29	6857924317348152962196438759681243
Element 30	8431659272913786545672491387845932
Element 31	2846357199578124363167498528352619
Element 32	7396241588619352745241789632184965
Element 33	9458362716729518431384725695136984
Element 34	4629317581857426397938654218492531
Element 35	2138459679742168535687392148274531
Element 36	7534189266817923452493658173982715
Element 37	2683459175146798239378215468219367

If you have any questions please contact me.