

# Simulation Software for Smart-Fridge and Sudoku Solver

by  
Mariusz, Jonas and Sreenivas

# Overview

- Simulation software for rendering images
- Smart-Fridge: Fruit(s) with varying stages of decay
- Sudoku Solver: Sudoku puzzles with various parameters

# Requirements

Feature sets:

- Sudoku Solver:
  - Difficulties: easy, medium and hard
  - Completion stages: unsolved, partially solved and fully solved
  - Errors: Apperance of intended errors in puzzle
  - Handwritten fonts: Various handwritten fonts and mixtures of them
  - Transformations
  - Noise: Lighting, coloring

# Requirements

Feature sets:

- Smart-Fridge:
  - Fruits: Banana and tomato model
  - Stages of decay: fresh, neutral and rotten
  - Number of fruits

# Requirements

- GUI:
  - User friendly
  - Clearly structured User Interface
- Performance:
  - Non realtime
- Environment:
  - Linux
  - Blender
- Reliability:
  - Groundtruth labels
- Version control and License: GIT, GNU General Public License v2.0
- Hardware:

## Minimum (basic usage) hardware

- 32-bit dual core 2Ghz CPU with SSE2 support.
- 2 GB RAM
- 24 bits 1280×768 display
- Mouse or trackpad
- OpenGL 2.1 compatible graphics with 512 MB RAM

## Recommended hardware

- 64-bit quad core CPU
- 8 GB RAM
- Full HD display with 24 bit color
- Three button mouse
- OpenGL 3.2 compatible graphics with 2 GB RAM

## Optimal (production-grade) hardware

- 64-bit eight core CPU
- 16 GB RAM
- Two full HD displays with 24 bit color
- Three button mouse and graphics tablet
- Dual OpenGL 3.2 compatible graphics cards with 4 GB RAM

# Software Lifecycle

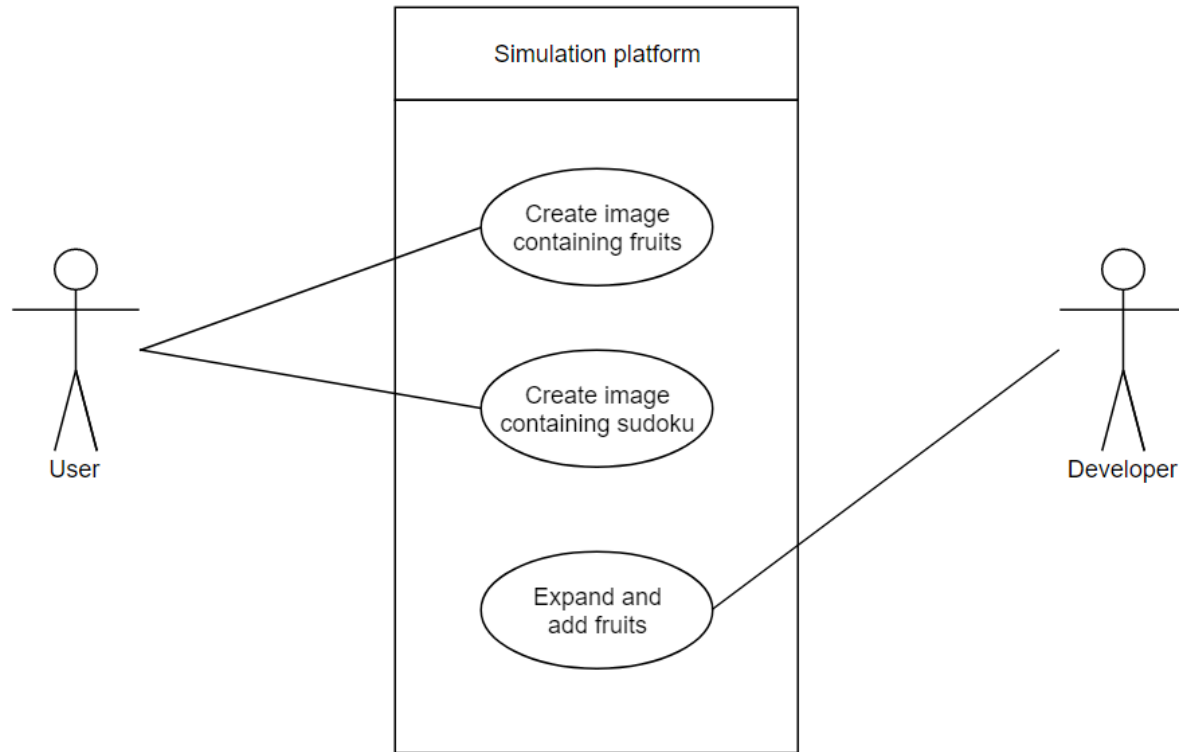
- Evolutionary Lifecycle

Selection reasoning:

- Addresses risks early (Risk choice similar to Spiral Model)
  - Changing requirements
  - Useful feedback loops  $\longleftrightarrow$  Customer satisfaction
- Fighting timelines: **fit to features/quality??** → exam deadline

	Risk mgmt.	Quality/ cost ctrl.	Predict-ability	Visibility of progress	Customer involvement
Code-and-fix	1	1	1	3	2
Waterfall	2	4	3	1	2
Spiral	5	5	3	3	3
Evolutionary prototyping	3	3	2	5	5
Staged delivery	3	5	3	3	4
Fit-to-schedule	4	3	5	3	2

# Use cases



# Use cases

<b>Number</b>	1	
<b>Name</b>	Rendering images for SmartFridge	
<b>Description</b>	A User goes through the dialogues to create an image containing various fruits of different rotten and non-rotten states.	
<b>Priority</b>	5	
<b>Preconditions</b>	Program was started	
<b>Postconditions</b>	None	
<b>Primary Actor(s)</b>	User	
<b>Secondary Actor(s)</b>	None	
<b>Trigger</b>	User chooses SmartFridge after program start.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Program shows start dialogue containing buttons for choice between SmartFridge and Sudoku
	2	User chooses SmartFridge
	3	Program displays parameter selection with default values
	4	User changes parameters at will
	5	Program displays rendered image for given input parameters
	6	User chooses whether to reject or save the image
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	4.1	User insert number and sort of fruit.
	4.2	Program shows preview of the scene to render
<b>Issues</b>	How does the selection look like? Does the user click through multiple images or does he get only one image per generation?	



# Use cases

Number	2	
Name	Rendering images for Sudoku Solver	
Description	A User goes through the dialogues to create an image dataset containing various Sudoku puzzles.	
Priority	4	
Preconditions	Program was started	
Postconditions	None	
Primary Actor(s)	User	
Secondary Actor(s)	None	
Trigger	User chooses Sudoku after program start.	
Main Scenario	<b>Step</b>	<b>Action</b>
	1	Program shows start dialogue containing buttons for choice between SmartFridge and Sudoku
	2	User chooses Sudoku
	3	Program displays parameter selection with default values
	4	User changes parameters at will
	5	Program displays rendered image for given input parameters
	6	User chooses whether to reject or save image
Extensions	<b>Step</b>	<b>Branching Action</b>
	4.1	Program shows preview of the Sudoku scene to render.
Issues	How does the selection look like? Does the user click through multiple images or does he get only one image per generation?	

# Use cases

Number	3	
Name	Add a fruit to the SmartFridge (Experimental Use case for scalability in future	
Description	A Developer goes through the dialogues to create an image containing one Sudoku.	
Priority	2	
Preconditions	Developer has information and access to plugin interface	
Postconditions	Program is still executable	
Primary Actor(s)	Developer	
Secondary Actor(s)	None	
Trigger	None	
Main Scenario	<b>Step</b>	<b>Action</b>
	1	Developer adds a blender compatible mesh for the new fruit
	2	Developer adds a colormap for the new fruit
Extensions	<b>Step</b>	<b>Branching Action</b>
	1.1	The developer has to apply the parameter types and names for his mesh generating routine

# Architecture

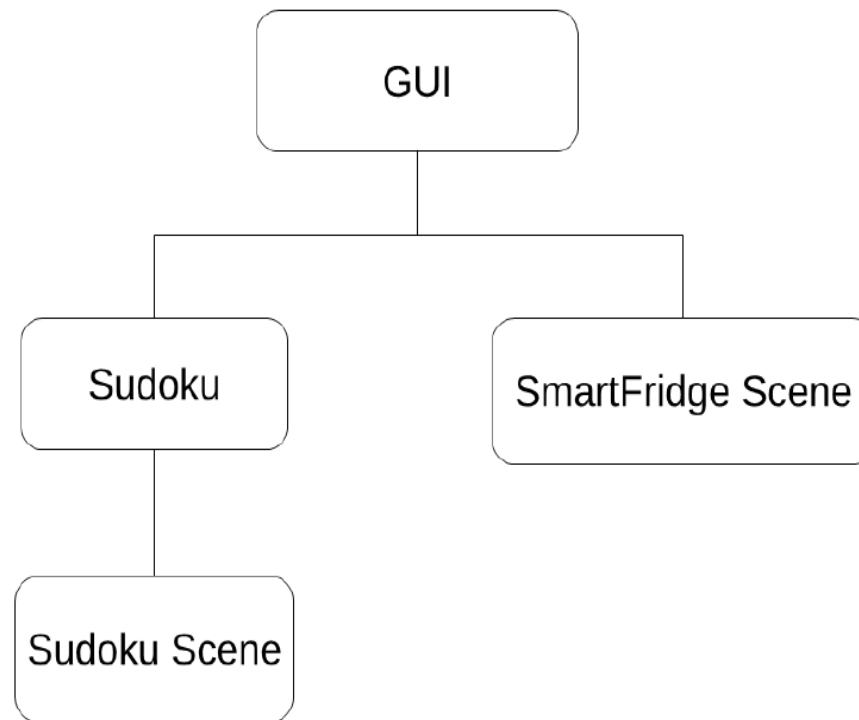
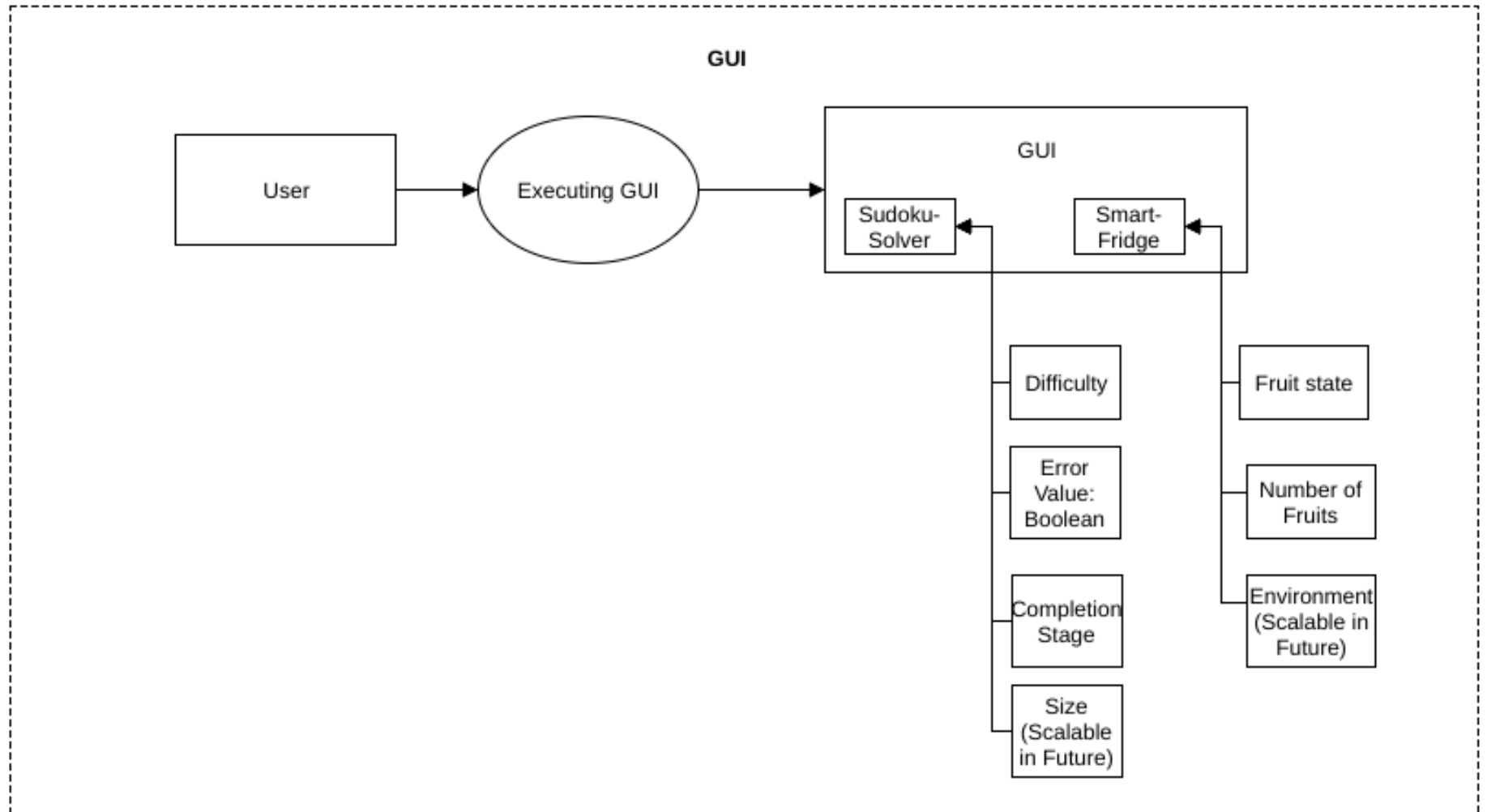
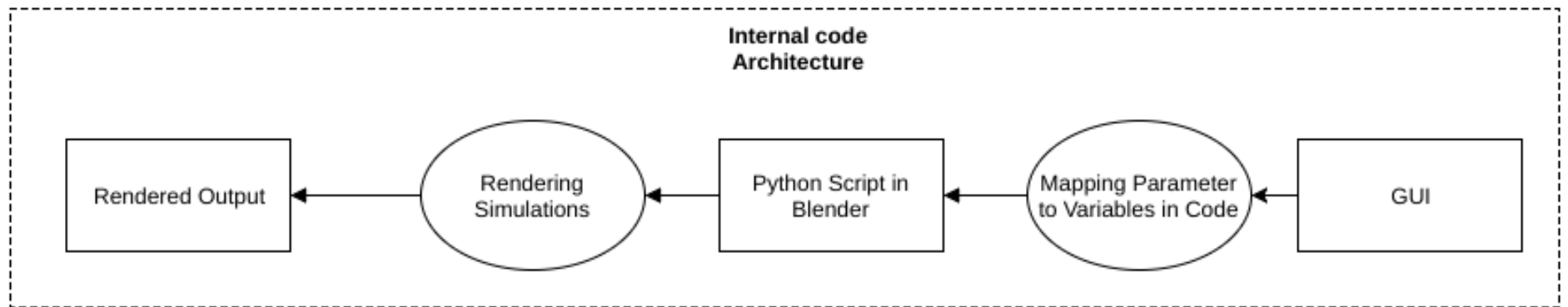


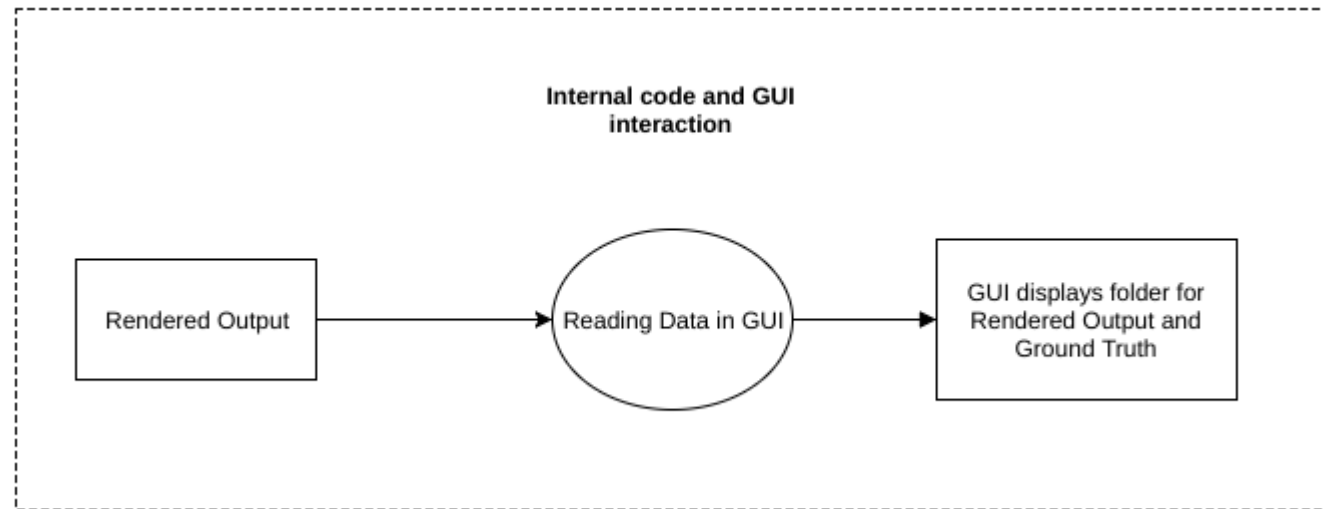
Figure 1: System architecture

The individual modules at the top layer include the GUI containing the choice between SmartFridge and Sudoku solver and the corresponding modules to produce the images. This Architecture is shown in Figure 1. The extra Sudoku class had to be implemented to conform the requirement for keeping the puzzles numbers while other parameters change. The blender script would have initialized them different numbers on every call.

# Workflow







# Joel Test

1. Do you use source control? → **git**
2. Can you make a build in one step? → **every module does**
3. Do you make daily builds? → **for every module**
4. Do you have a bug database? → **No**
5. Do you fix bugs before writing new code? → **Yes**
6. Do you have an up-to-date schedule? → **Yes**
7. Do you have a spec? → **Yes**
8. Do you do hallway usability testing? → **No**

# Testing, Review & Refactoring

- Testing
  - We Unit tested and Blackbox-tested each other's code
- Reviews
  - We had peer reviews with each other
- Refactoring
  - No, since we're not maintaining yet



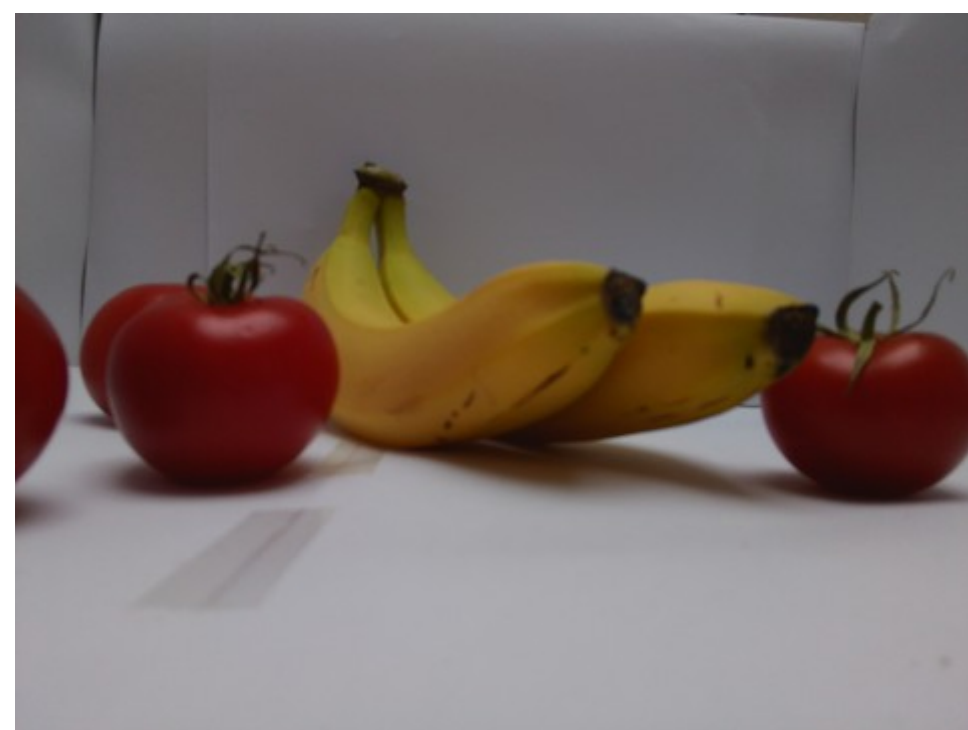








# Real vs. rendered

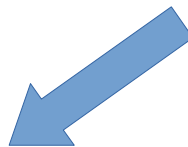




6				2	7	9		5
5		9	6		1		2	7
			9	8		6	3	1
	5		7				9	
			1					4
	4		8	5			1	
4	8		2		6	1	5	
	9	5	4		8	7	6	
		6	5	1	9	4	8	



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

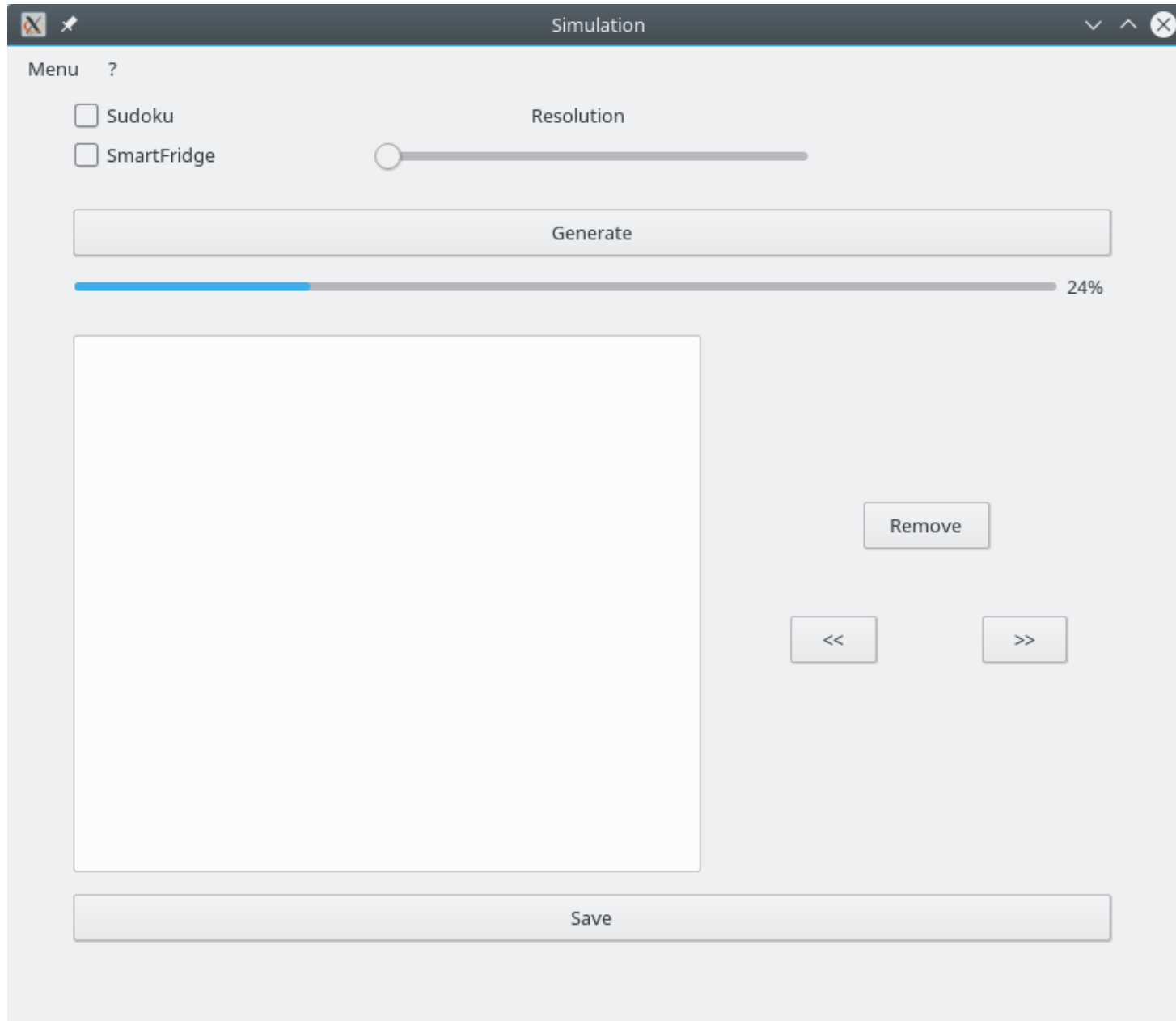


6	1	8	3	2	7	9	4	5
5	3	9	6	4	1	8	2	7
7	2	4	9	8	5	6	3	1
3	5	1	7	6	4	2	9	8
8	6	2	1	9	3	5	7	4
9	4	7	8	5	2	3	1	6
4	8	3	2	7	6	1	5	9
1	9	5	4	3	8	7	6	2
2	7	6	5	1	9	4	8	3



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

# GUI Version 1.0



# GUI Version 2.0

Form

Frontend for Sudoku Solver input images simulation with a lot of good looking sliders.

Edit generated database

Numbers

Printed

TextLabel

TextLabel

TextLabel

TextLabel

TextLabel

TextLabel

Difficulty

Low

1

TextLabel

TextLabel

TextLabel

TextLabel

TextLabel

Number of images

TextLabel

TextLabel

TextLabel

TextLabel

TextLabel

TextLabel

TextLabel

Generate

24%

Save



# GUI Version 3.0

SudokuSolver

FileAbout

Standard view

Path to the database ?

Open

Resolution ?

300

x

300

Difficulty ?

☐ 0

☐ 1

☐ 2

☐ 3

Errors ?

☐ yes

☐ no

Completion stage ?

☐ solved

☒ partially solved

☐ not solved

Number of images ?

1

Expert view

Show preview

Generate

# Summary

- Simulation program for rendering Sudoku Puzzles and fruits for Smart-Fridge
- Modular with three different modules: Smart-fridge simulation, Sudouku simulations, GUI
- Evolutionary Life-cycle followed