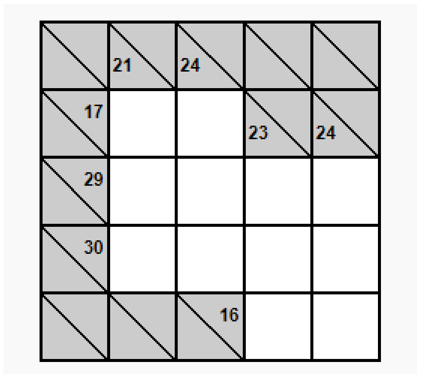
Requirements Document

Team PI-B

9 February 2020



Kakuro

Team members

Name	ID Number
Sajib Ahmed	A
Yaroslav Bilodid	B
Jesse Desmarais	C
Antoine Farley	D
Marc Hegedus	E
Katerina Tambakis	F
Yingjie Zhou	G

1 System

1.1 Purpose

1.2 Context

1.3 Business Goals

2 Domain Concepts

2.1 Objectives

The project to be completed in COMP 354 of Winter 2020 is to create a functional replica of the puzzle game Kakuro. Team PIB's version in iteration one will consist of a singular solvable puzzle. Functionality will be limited to input from the client and feedback for the verification of a correct solution. The main objective is to apply software engineering techniques for the development process to be test-driven, agile, and object-oriented. There will be three iterations, each having their own deadlines. Efficient management and communication amongst our group is understood to be central in accomplishing the required tasks. The client wants the following as deliverables for the first iteration: a basic graphical user interface, a model-view-controller architecture coded in Java, and a categorized set of use cases.
The three iterations will each have a document to be handed in to the client. The information regarding their naming, deliverables, and dates are tabulated below:

Iteration	Deliverable	Date
Requirement	Requirements Document	2020/02/09
Design	Design Document	2020/03/15
Implementation	Final Document	2020/04/5

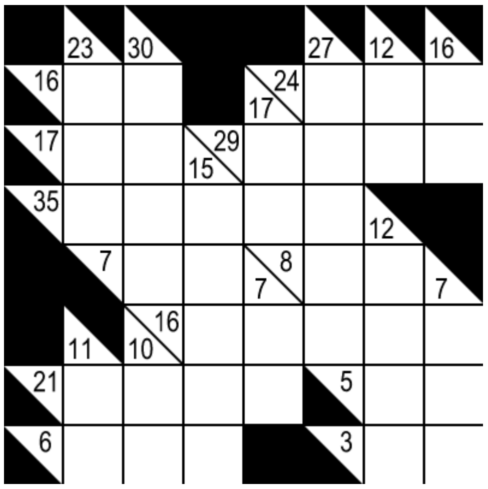
Project Timeline

2.1.1 Kakuro Puzzle Game Specifications

Kakuro is a japanese mathematical logic puzzle with rules similar to the crossword and sudoku. To develop a successful graphical user interface, the game's rules and characteristics are to be clearly detailed. This will allow clients and various actors to recognize the product for an intuitive feel. Due to the simplistic nature of the game, development will be focused on limiting user interaction and increasing response. This will be realized in the form of input validation. Users will interact with a single interface where numerals from one to nine are to be entered. If incorrect, a red translucent overlay will be superimposed upon the respective tile. A thorough description of Kakuro's mechanics is provided below.

2.1.1.1 Gameboard

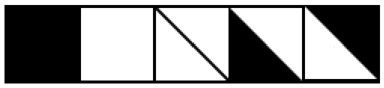
Kakuro's classic form is composed of a monochromatic square matrix where three different tiles are displayed. They are the following: blank, input, and clue. Blank tiles are black and cannot contain any value. Clients are to interact only with the input tile. Numerical values will be inserted to complete the sums found within the clue tiles. Clue tiles are bisected diagonally where the lower and higher sums must correspond to the addition of all the vertically and horizontally aligned numerals in the input tiles, respectively. Input tiles along a specified direction can only have unique values from 1 to 9. Their intersections must then contain the same numeral. As such, a solution is obtained when every sum has been rightfully achieved throughout the gameboard matrix. The figure below shows a Kakuro puzzle at the easy difficulty.



Easy Kakuro gameboard

2.1.1.2 Tiles

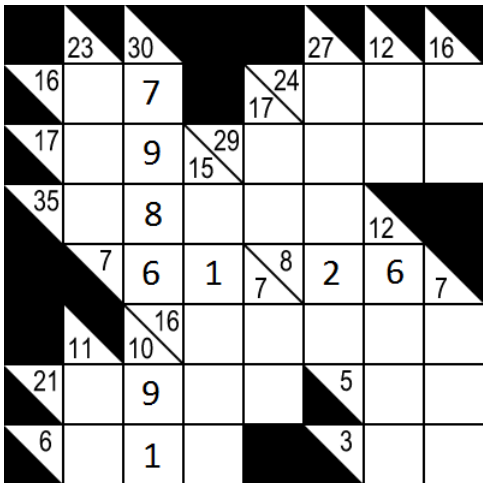
The three distinct types of tiles are shown in the figure below in the given order: blank, input, and clue. Blank tiles are always filled with black and no numeral is ever shown within its confines. Input tiles are white and users are to fill them with a number from one to nine. Clue tiles are bisected from the top left to the bottom right and come in three separate formats. If a corner is white a sum will shown, whereas if it's black none will appear. As such, both corners may be hold a sum, or only one of the two might.



Tile variation as seen in Kakuro

2.1.1.3 Sum and intersections

A vertical and horizontal column has been filled out with a possible solution to the gameboard shown in section 2.1.1.1. The top of the vertical column suggests a sum of 30 is needed from the input tiles vertically below. A combination of 9-8-7-6 must be placed within the four allotted slots. This vertical column had two clues where the second formed a break. Clue tiles that appear sequentially along a particular column represent a new sum to acquire from the input tiles. The second reads a sum of 10. A combination of 9-1, 8-2, 7-3, or 6-4 is allowed to be placed. The horizontal that has been filled shares a tile with the vertical column. This is called an intersection. The number in that input tile must be correct for both columns. The horizontal's first number is 7. Since a there lies a 6, only a 1 can be placed within the adjacent tile. The flow of the game concerns itself by correctly forming the proper sums and accommodating the many intersections between columns.



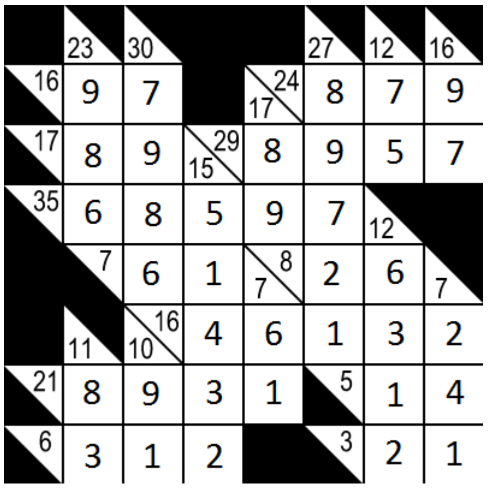
Partially completed Kakuro gameboard

2.1.1.4 Solution

The final solution to the gameboard shown in section 2.1.1.1 is given below. Each of the following requirements has been met:

- Numerals Numbers used within the input tiles are limited from 1 to 9.
- Sum Every input tile along a column corresponds to its vertical and/or horizontal clue tile.
- Intersection Every intersection between vertical and horizontal columns had a common numeral that worked towards the respective sum.
- Completion Every input tile has been correctly filled out in the gameboard matrix.

The difficulty is set to easy due to the clues being too telling of the numbers needing to be entered. Taking a look at the top left, there's only one variation of numbers which can be inserted. This forms a good base to complete the rest of the puzzle.



Kakuro gameboard solution

3 Actors

4 Use Cases

4.1 Overview

Figure 1: Use Case Diagram

4.1.1 Use Case 1

Name
Give a name.

Summary
A short summary/description/story.

Actors

Precondition

Main Scenario

1. Describe step 1.
2. Describe step 2.
3. Describe step 3.

Exceptions

Postcondition

Priority

Traces to Test Cases
Add when test cases done.

4.1.2 Use Case 2

5 Non-Functional Constraints

6 Data Dictionary

7 References

A Description of File Format: Tasks

Describe input file format.

B Description of File Format: Persons

Describe output file format.