CS101 Project 2015

Mathematics Scrabble

Team ID: 407

Naman Gupta(TL)	140040057
Aditya Pratap	140040034
Debarnab Mitra	140070037
Rhythm Jain	140040054

Problem Statement

- The aim of the project is to build a software version of the game with basic user interface.
- The game will be having Artificial Intelligence to play as opponent and scoring function along with equation building and identification functions.

Task Specifications

1. A randomising function that distributes random tiles to players.
2. To build a set of functions which will identify as well as verify the equation built on the board.
3. Building a permutation function that will enable the computer to build equations on its own. This is required because the player plays against the computer.
4. Building a scoring function which can award score based on the laid rules of scoring.
5. Providing decent graphics for smooth gameplay.

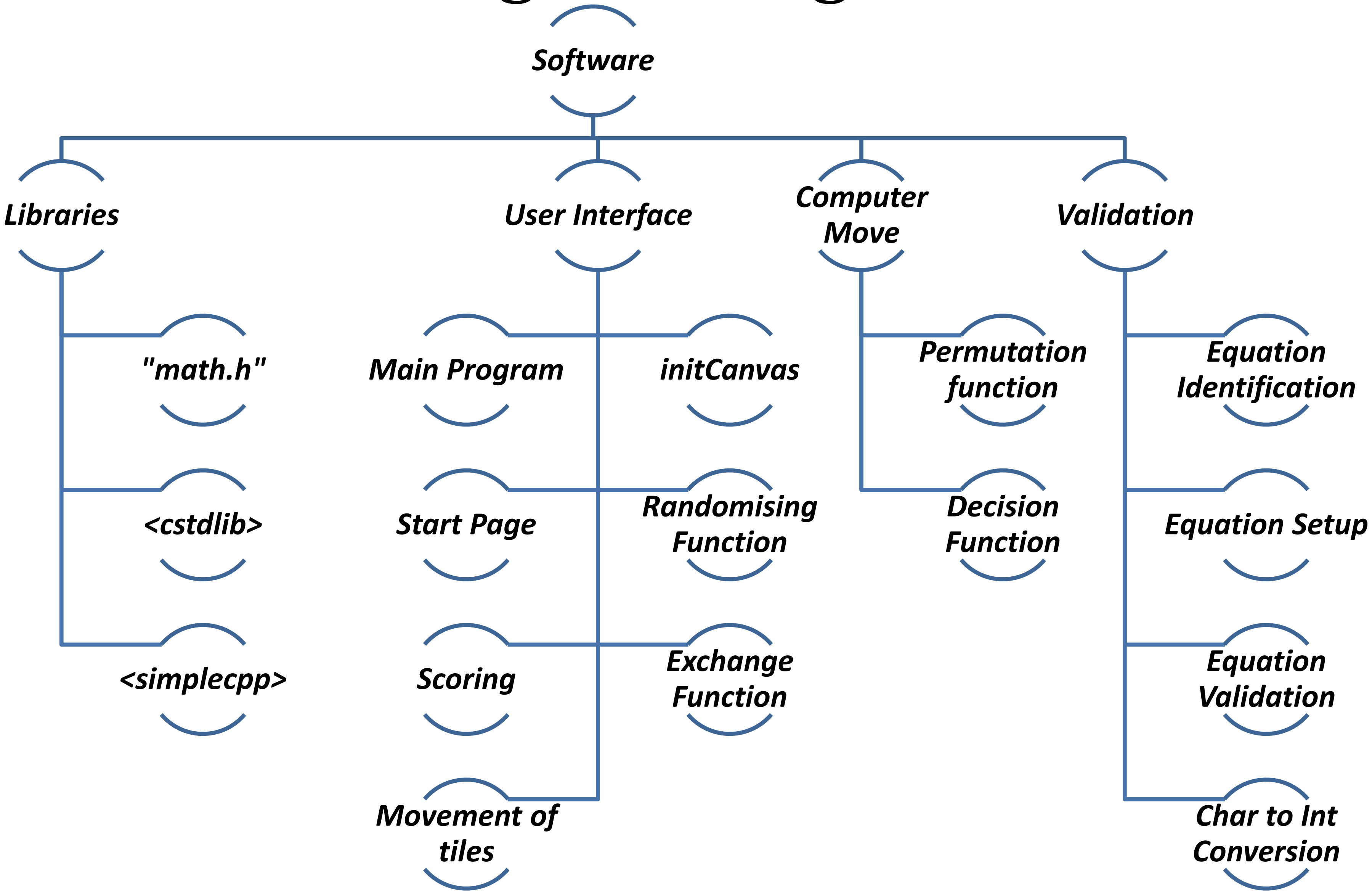
Project Plan

The Project has been completed in two Stages:

1. Developing necessary functions and codes to support the functioning of the game. Developing Interface and basic Artificial Intelligence.
2. Integrating all the programs and developing the Artificial Intelligence further to produce better equations. Improving Graphics and functions to enhance gaming experience.

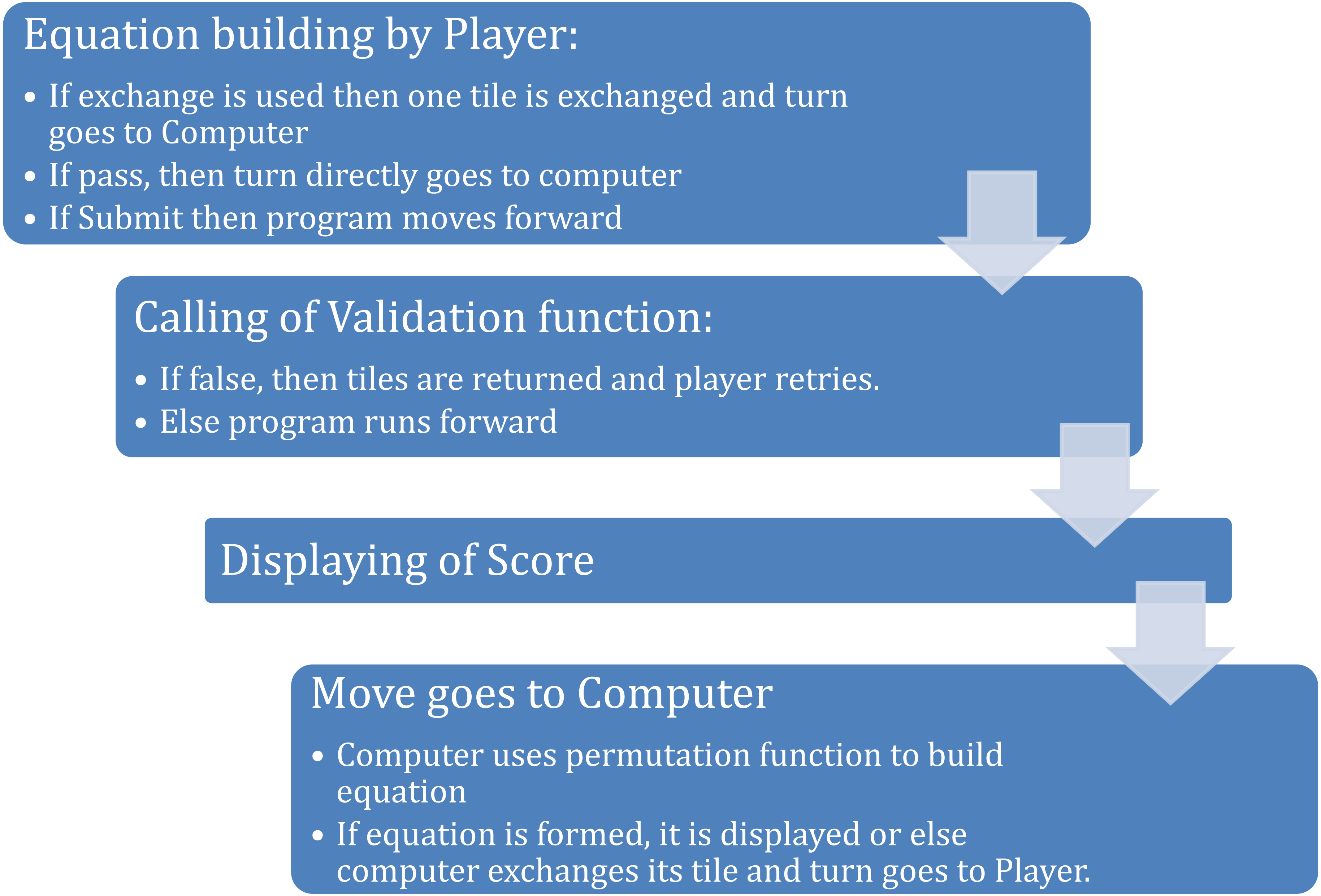
- The first Goal was achieved in the second week itself
- By third week we were able to run the game without the Artificial Intelligence i.e. UI integration was finished.
- Most of the second goal was finished by the Fourth and final week.

Program Diagram



- Software
  - Libraries
    - "math.h"
    - <cstdlib>
    - <simplecpp>
  - User Interface
    - Main Program
    - initCanvas
    - Start Page
    - Randomising Function
    - Scoring
    - Exchange Function
    - Movement of tiles
  - Computer Move
    - Permutation function
    - Decision Function
  - Validation
    - Equation Identification
    - Equation Setup
    - Equation Validation
    - Char to Int Conversion

System Diagram



Equation building by Player:

- If exchange is used then one tile is exchanged and turn goes to Computer
- If pass, then turn directly goes to computer
- If Submit then program moves forward

Calling of Validation function:

- If false, then tiles are returned and player retries.
- Else program runs forward

Displaying of Score

Move goes to Computer

- Computer uses permutation function to build equation
- If equation is formed, it is displayed or else computer exchanges its tile and turn goes to Player.

Innovations and Challenges

- The game demanded that players don’t see each others tiles, hence we had to develop artificial intelligence to support the game.
- Since the game involved usage of operators as well as integers. We couldn’t use ‘int’ or ‘float’, so we used ‘char’. But that caused new problem of validating equations from ‘char’.
- The program has functions that can build numbers and equations out of ‘char’ array. This uses ‘identify()’ and ‘char_to_int()’ functions. These functions can be used to build some library in future.
- The Artificial Intelligence was a challenge too. To make it time efficient was tricky. We had to re-build our permutation function and consider special cases to make it time efficient.

Tasks Completed

- For task 1, we planned on using srand() function but that always returned the same random values and hence we used “srand(time(NULL))”.
- Task 2 was tricky, since it demanded building equations out of ‘char’ array. This was done using multiple set of ‘bool’ functions.
- Task 3 was a long and delicate code and was to be built carefully. Also at first it was not that efficient and we had to make changes to make it more effective and efficient.

Tasks Completed

- Task 4 was completed successfully without much difficulties.
- Task 5 was one of the most important tasks. Since we used initCanvas, we had limited resources. We faced some problems like unwanted disappearing of tiles when two initCanvas windows were opened. Later we resolved this by using single window.

Testing

- The code was tested at various levels using test data.
- The test data included using ‘test main programs’ where artificial input were supplied from our side to check the functioning of code.
- At various places ‘if’ conditions were verified using ‘cout’ commands in the terminal window.
- The testing was performed successfully.

Performance Metrics

- The graphics library used i.e. initCanvas, due to its limitations causes the game to slow down. The displaying of new tiles takes time.
- Also the permutation function may take time to build a valid equation.
- The validation functions and Computer moves run smoothly.

Re-usability features

- We have used GNU style indention.
- We have used ‘Delimiter-separated Words’ for naming convention.
- The functions have added comments to help others understand the code.
- The set of functions for validation can be used to build a separate library for verifying equations in a string. These functions and permutation function can be reused for any other program.

Future Enhancements

- The game can be developed with better graphics for interactive educational purposes. This software can then be used in schools or with kids of age < 10.
- Multiplayer version of the game using LAN connections can be built so two players can play the game simultaneously. This will requires access to LAN.
- Including other mathematical operators like square root, square etc. This will add fun to the game. Also this game can be launched on an open platform.