An overview on project Gamebox in C++ (version 1.000)

by-

Name: Md. Tasnimul Khair Tousif

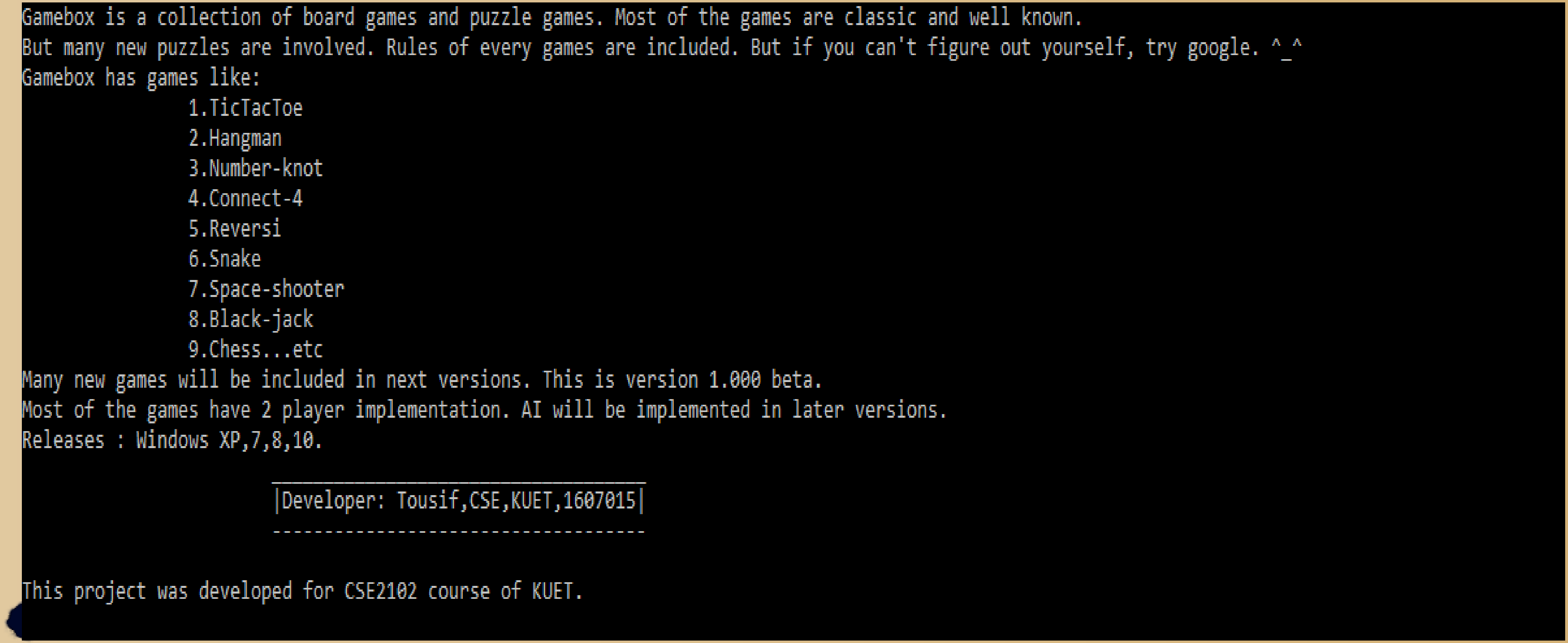
Roll: 1607015

Sec: A

Department of CSE,

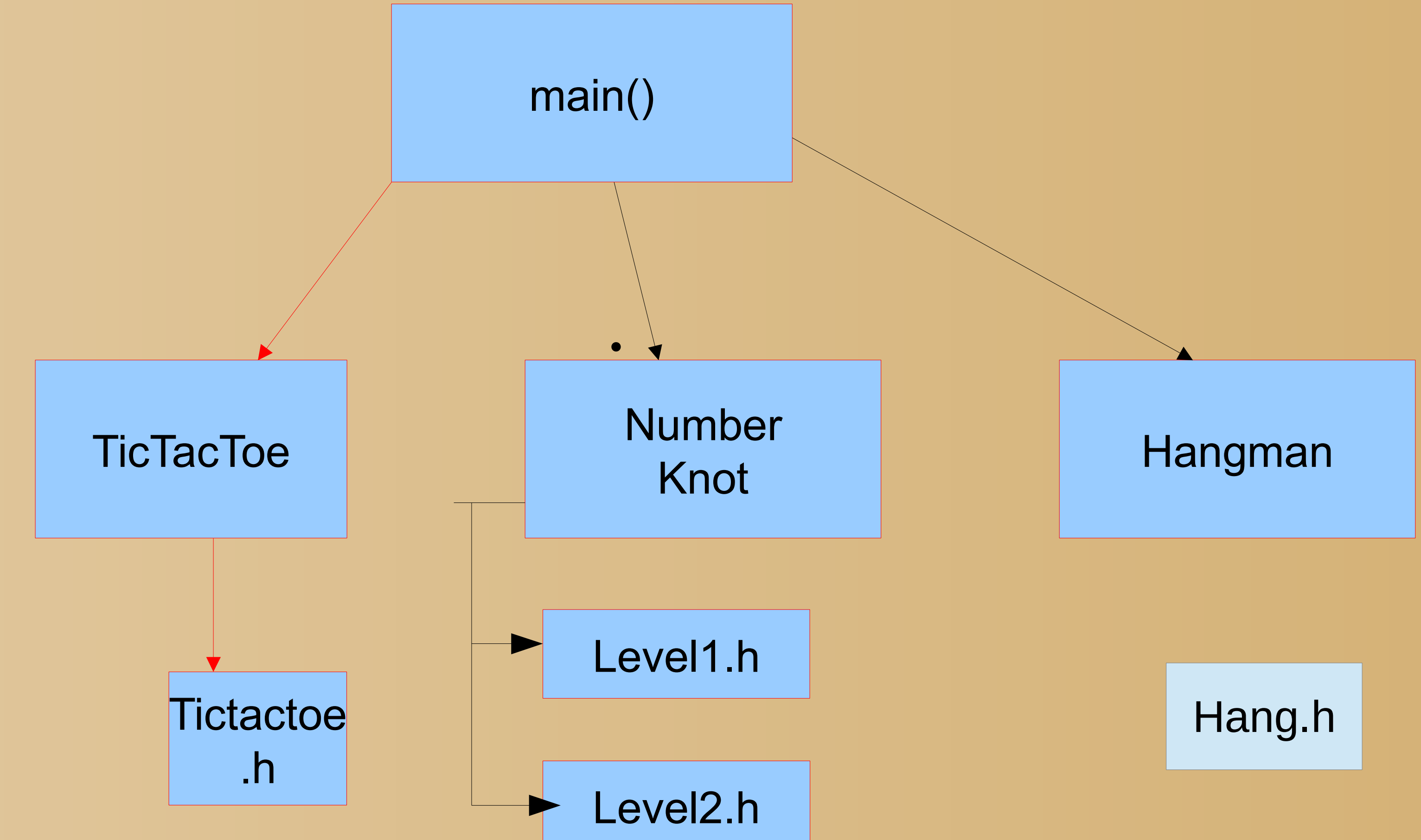
Khulna University of Engineering and Technology, Khulna

Gamebox: intro

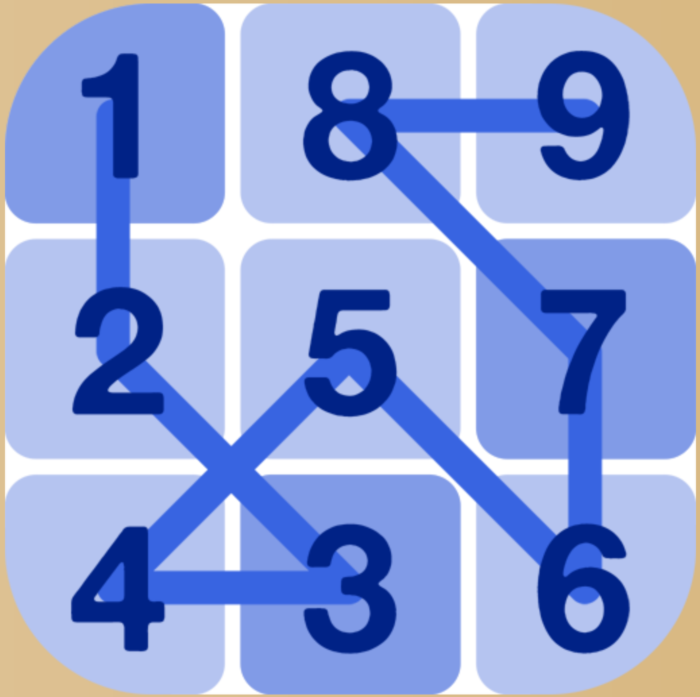


```
Gamebox is a collection of board games and puzzle games. Most of the games are classic and well known.
But many new puzzles are involved. Rules of every games are included. But if you can't figure out yourself, try google. ^_^
Gamebox has games like:
                1.TicTacToe
                2.Hangman
                3.Number-knot
                4.Connect-4
                5.Reversi
                6.Snake
                7.Space-shooter
                8.Black-jack
                9.Chess...etc
Many new games will be included in next versions. This is version 1.000 beta.
Most of the games have 2 player implementation. AI will be implemented in later versions.
Releases : Windows XP,7,8,10.

                        ____________________________________
                        |Developer: Tousif,CSE,KUET,1607015|
                        ------------------------------------


This project was developed for CSE2102 course of KUET.
```

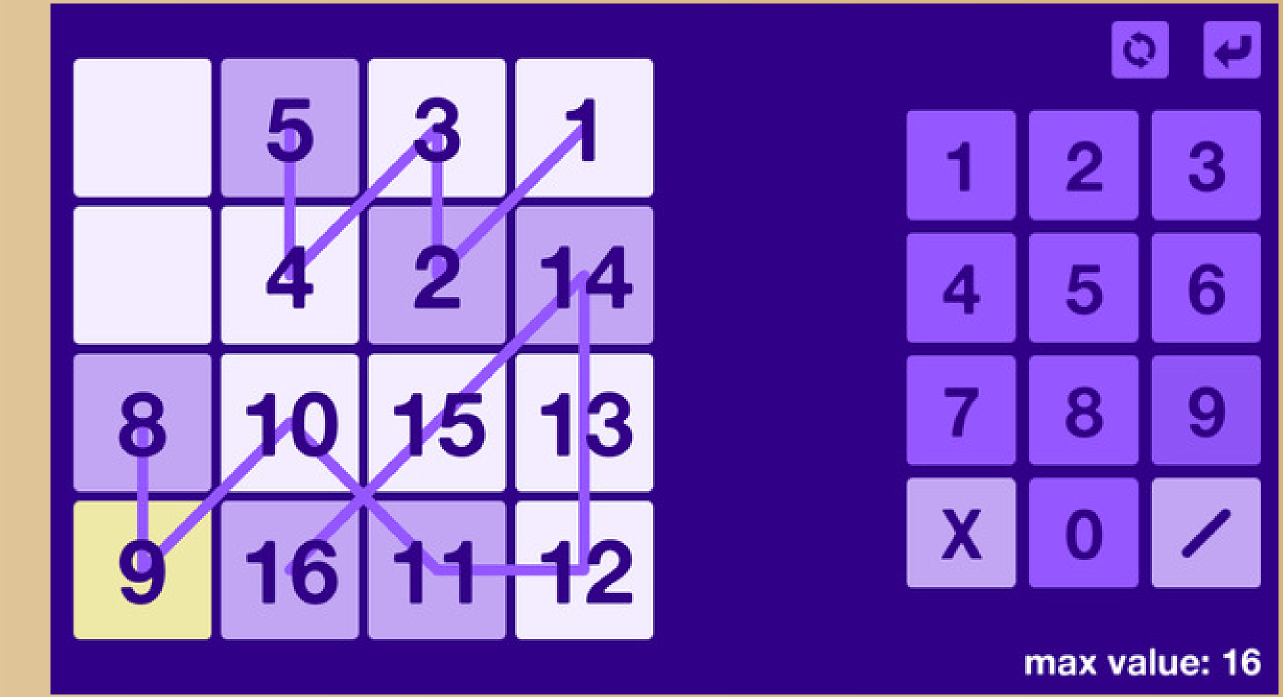
A General Map



Overview: Number Knot



Overview: Number Knot

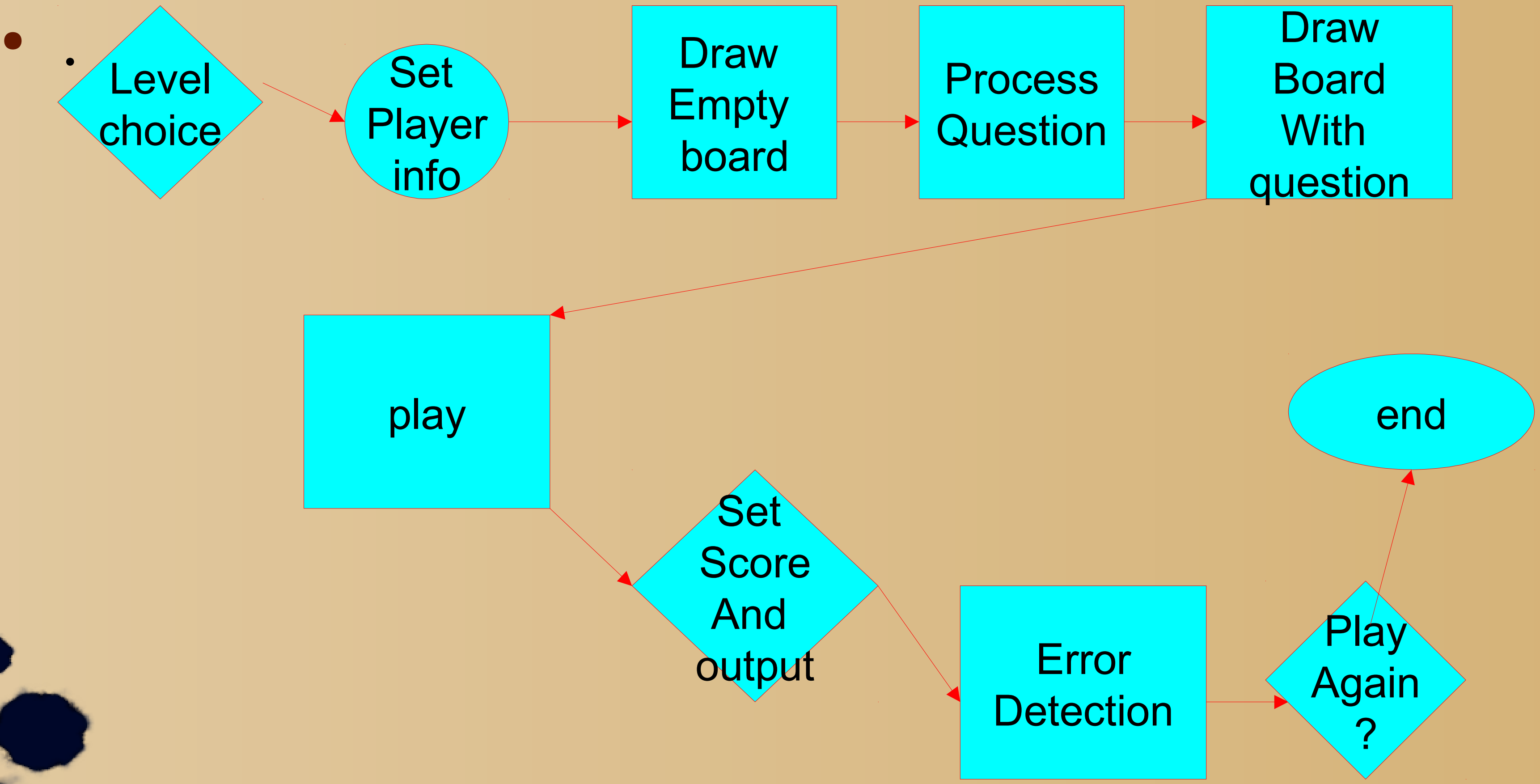


	5	3	1
	4	2	14
8	10	15	13
9	16	11	12

max value: 16

Flow of control: Number Knot

- .



Level choice

Set Player info

Draw Empty board

Process Question

Draw Board With question

play

Set Score And output

Error Detection

Play Again ?

end

Flow of Control: play() (number knot)

The play() function does these-

1.while(moves != 0):

2.input();

3.if (exit_status() == 1) break loop;

4.end of loop.

5.calculate score and output.

6.preprocess answer and output numeric and texual representation of answer.

Flow of control: input() (number-knot)

Input() is invoked in play(). Does these.

1.check row and column input also check validity

2.if valid, mask_board[r][c] = true (denotes input is taken)

3.take number as input. Also check validity.

4.if valid insert to temp_sol_board[r][c], if val == standard_sol_board[r][c]: process and draw temp_board().

Error detection(number-knot)

- is_exit() == true

1	2(q)	3
8(q)	9	4
7	6(q)	5

- solution_board

1	2(q)	5
8(q)	9	4
9	6(q)	5

- main_board

		–
–		

temp_board

exit_status(number-knot)

This function is useful to force exit when all boxes are filled.

- is_exit() == true

_	q	_
q		_
	q	

- is_exit() == false

- If all grids from main board are free of ‘_’ then exit is possible.

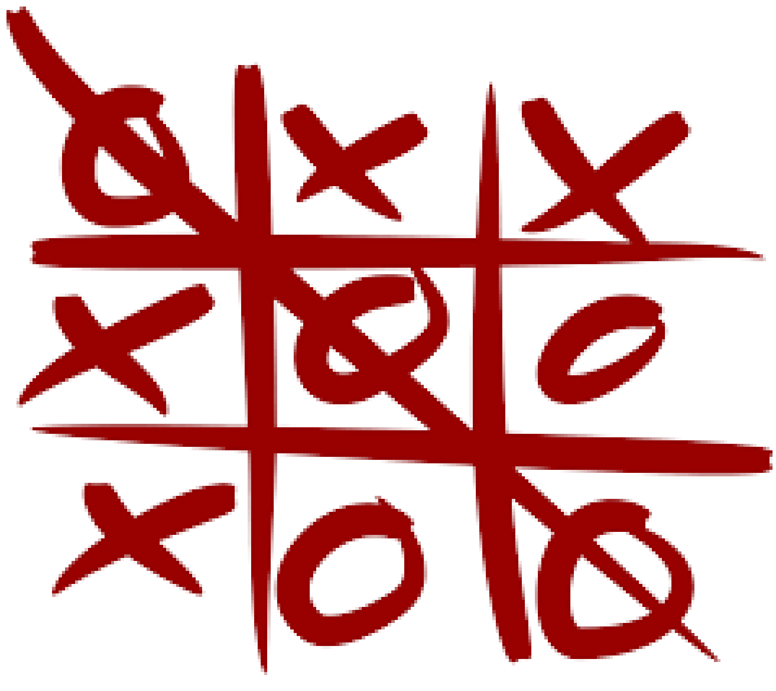
Check input validity: mumber-knot

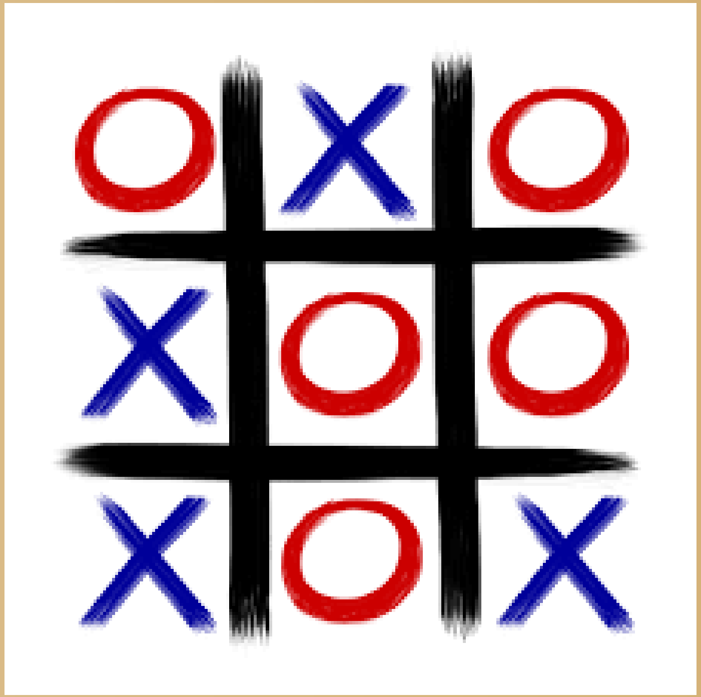
5(q)		1(q)
4	3	2(q)

f	f	f
T_q	f	T_q
T	T	T_q

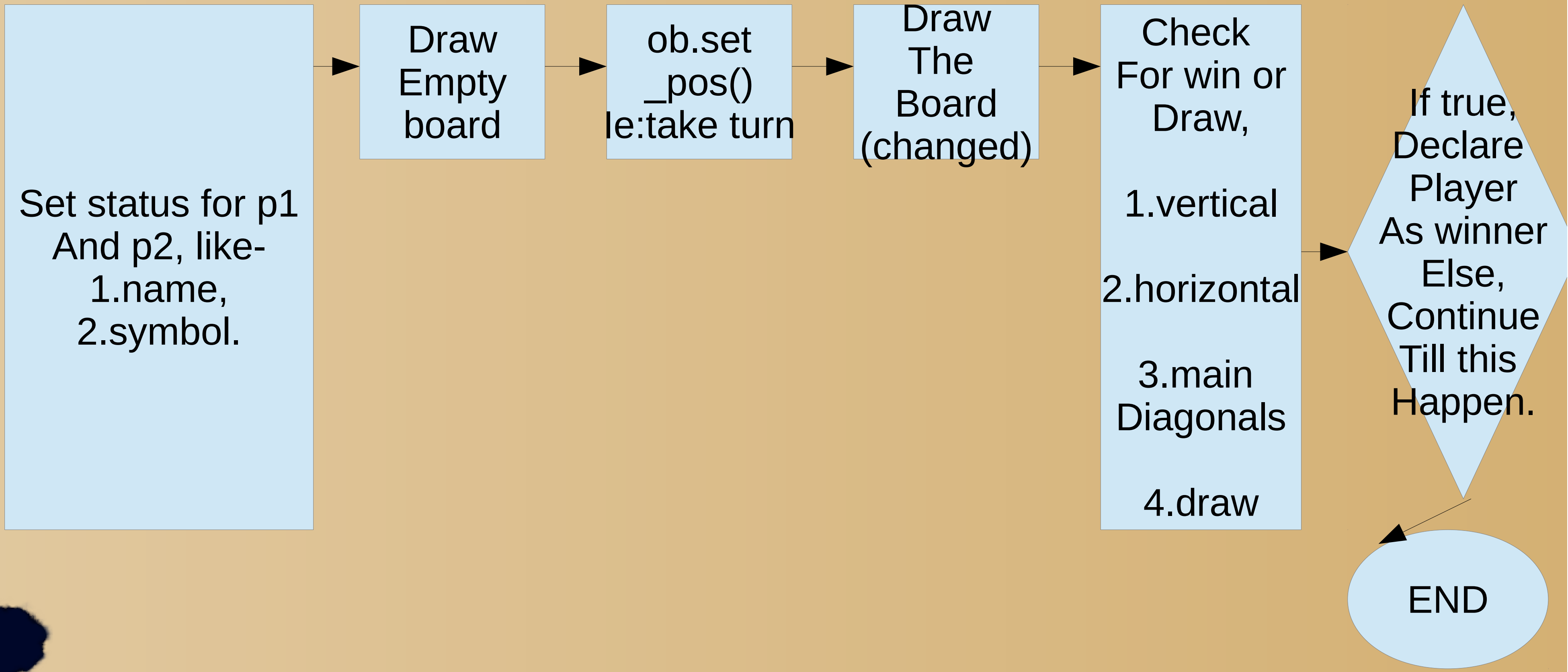
Where mask[r][c]==false, its ok to input.

Overview: TicTacToe





Flow of control: TicTacToe



Set status for p1
And p2, like-
1.name,
2.symbol.

Draw
Empty
board

ob.set
_pos()
le:take turn

Draw
The
Board
(changed)

Check
For win or
Draw,

1.vertical

2.horizontal

3.main
Diagonals

4.draw

If true,
Declare
Player
As winner
Else,
Continue
Till this
Happen.

END

Mechanism: TicTacToe

- Elements of 2D array are accessed via 2 for loops.
- Loop body looks like this:
- for(int i=0; i<x; i++)
- {for(int j=0; j<x; j++)
- {}}
- For horizontal – a[i][j]
- For vertical – a[j][i]
- For diagonal(main) – a[i][i] //only one loop
- For other diagonal(main) – use while loop setting , I = 0 and J = dimension. Keep looping till I<dim && J>-1

Overview: Hangman

Overview: Hangman

- Pseudocode:

```
Create a group of words.
Pick a random word from the group as the secret word.
While player hasn’t made too many incorrect guesses and hasn’t guessed the secret word:
Tell player how many incorrect guesses he or she has left.
Show player the letters he or she has guessed.
```

Overview:Hangman

- Pseudocode(continued):

```
Show player how much of the secret word he or she has guessed.
Get player’s next guess.
While player has entered a letter that he or she has already guessed
Get player’s guess
Add the new guess to the group of used letters
If the guess is in the secret word
Tell the player the guess is correct
Update the word guessed so far with the new letter
```

Overview: Hangman

- Pseudocode(continued):

```
Otherwise
Tell the player the guess is incorrect.
Increment the number of incorrect guesses the player has made.
If the player has made too many incorrect guesses:
Tell the player that he or she has been hanged.
Otherwise:
Congratulate the player on guessing the secret word.
```

Features used

1.Namespace: was used in hang.h(line 23), tictactoe.h(line 187), number_knot.h(line 8, encapsulation of level 1 & 2). In order to provide naming collisions of introduction(), getFileContents(ifstream&), print_ascii_board() functions. It was really useful.

2.Getter-Setter functions: was used in level_1.h, level_2.h(line 12)-set; tictactoe.h(line 44)-set, player.h(line 14-set, line 21-set, line 30-get), hang.h(line 102)-set.

Features used

3.Formating I/O: used in tictactoe.h, in tic::introduction() (line 227,230). Also in number_knot.h knot::introduction() (line 46,48). Needed <iomanip> header.

4.File: in main.cpp – line 36,58_ascii_art(), in number_knot.h line 31,9; in level_1.h,level_2.h line 303, 288; in tictactoe.h (in many places to denote state of the game in a file.) Files were opened in append mode for state documentation.

- Ex:ofstream file1("tictactoe_log.txt",std::ios_base::app);

Features used

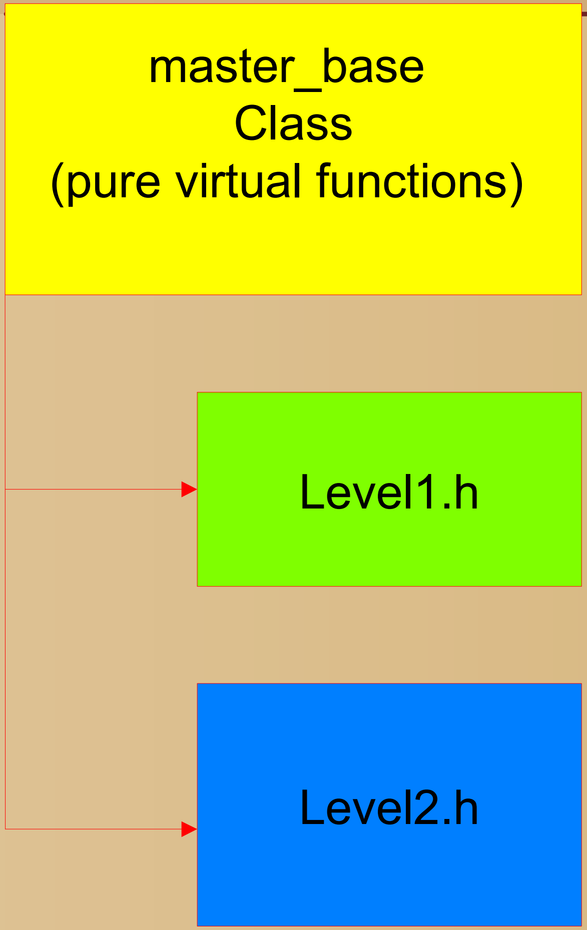
- 5.Class, objects, constructor, destructor: All programs were designed in an OOP manner, so they were used abundantly.
- 6.Operator Overloading: In tictactoe.h (line 27,171) used outpout<< operator overloading method to draw game board. And used throughout the program.
- 7.getline(): main.cpp (line 45), used in getFileContents (ifstream& File) function to read ascii lines fully till “\n”.

Features used

- 9.Template versions of STL member functions were used in hang.h (line 113-120=pair, line 139=static_cast<>).
- 10.STL vector,map and random_shuffle algo was used in hang.h (line 140,19,20). Many .h used string container class.
- 11.Functor used in level_1.h (line 304) and Lamda expression was used in level_2.h (line 289) for score evaluation.
- 12.Inheritance was used to design levels from a master base abstract class in number-knot game. Master.h has pure virtual function to force inherited classes to have its own implementations. New functionalities added.

Inheritance Map of Number Knot

- .



master_base
Class
(pure virtual functions)

Level1.h

Level2.h

Features used

- 13.catch,try mechanism was used for error handling in hang.h (line 229 to 240).
- 14. To handle invalid inputs, input_as_int() function was written. (level_1.h line 178) Also invalid array indexing was not permitted.
- 15. ascii art was found here: http://patorjk.com/software/taag/#p=display&h=0&v=0&f=Banner&t=NUMBER-KNOT

  Drawing from a file was found here:

  http://www.cplusplus.com/forum/general/58945/
- https://stackoverflow.com/questions/18728754/checking-input-value-is-an-integer

End