

CHESS

Team ID: 181

Hritik Singh, 140010062 (Team Leader)

Sushrut Phadke, 140010002

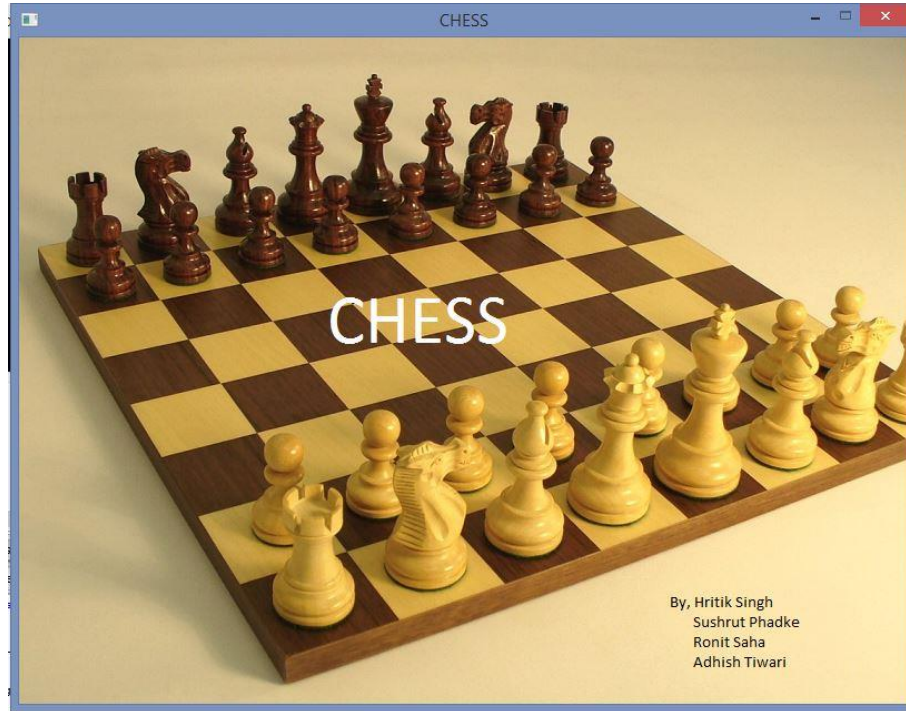
Ronit Saha, 140010039

Adhish Tiwari, 140110022

Problem Statement

1. Development of a two – player chess game that serves as an entertainment tool.
2. Making it as User friendly as possible.
3. Checking validity of moves of each piece.
4. Keeping record of pieces that each player has killed.
5. Displaying the winner at end of the game or telling the players if the game is a Draw.

Project Screenshots/Video



Screenshot: Initial Screen

```
"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
Welcome to 2-player Chess
Enter Player Names:
Whites: XYZ
Blacks:
```

```
"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
Welcome to 2-player Chess
Enter Player Names:
Whites: XYZ
Blacks: ZYX

      BLACKS
8 | r | n | b | q | k | b | n | r |
7 | p | p | p | p | p | p | p | p |
6 |   |   |   |   |   |   |   |   |
5 |   |   |   |   |   |   |   |   |
4 |   |   |   |   |   |   |   |   |
3 |   |   |   |   |   |   |   |   |
2 | P | P | P | P | P | P | P | P |
1 | R | N | B | Q | K | B | N | R |
  | a | b | c | d | e | f | g | h |

      WHITES

Killed pieces:
XYZ<White> plays:
Enter:-
Initial Position:
```

Screenshots: Basic Pawn Moves

```
"F:\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
XYZ<White> plays:
Enter:-
Initial Position: e2
Final Position: e4

      BLACKS
8 | r | n | b | q | k | b | n | r |
7 | p | p | p | p | p | p | p | p |
6 |   |   |   |   |   |   |   |   |
5 |   |   |   |   |   |   |   |   |
4 |   |   |   |   | P |   |   |   |
3 |   |   |   |   |   |   |   |   |
2 | P | P | P | P |   | P | P | P |
1 | R | N | B | Q | K | B | N | R |
  | a | b | c | d | e | f | g | h |

      WHITES
Killed pieces:
ZYX<Black> plays:
Enter:-
Initial Position:
```

```
"F:\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
ZYX<Black> plays:
Enter:-
Initial Position: a7
Final Position: a5

      BLACKS
8 | r | n | b | q | k | b | n | r |
7 |   | p | p | p | p | p | p | p |
6 |   |   |   |   |   |   |   |   |
5 | p |   |   |   |   |   |   |   |
4 |   |   |   |   | P |   |   |   |
3 |   |   |   |   |   |   |   |   |
2 | P | P | P | P |   | P | P | P |
1 | R | N | B | Q | K | B | N | R |
  | a | b | c | d | e | f | g | h |

      WHITES
Killed pieces:
XYZ<White> plays:
Enter:-
Initial Position:
```

Screenshots: Knight Moves

```
"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
XYZ<White> plays:
Enter:-
Initial Position: g1
Final Position: f3

      BLACKS
8  r  n  b  q  k  b  n  r
7  _  p  p  p  p  p  p
6  _  _  _  _  _  _  _
5  p  _  _  _  _  _  _
4  _  _  _  P  _  _  _
3  _  _  _  _  N  _  _
2  P  P  P  P  _  P  P
1  R  N  B  Q  K  B  R
   a  b  c  d  e  f  g  h

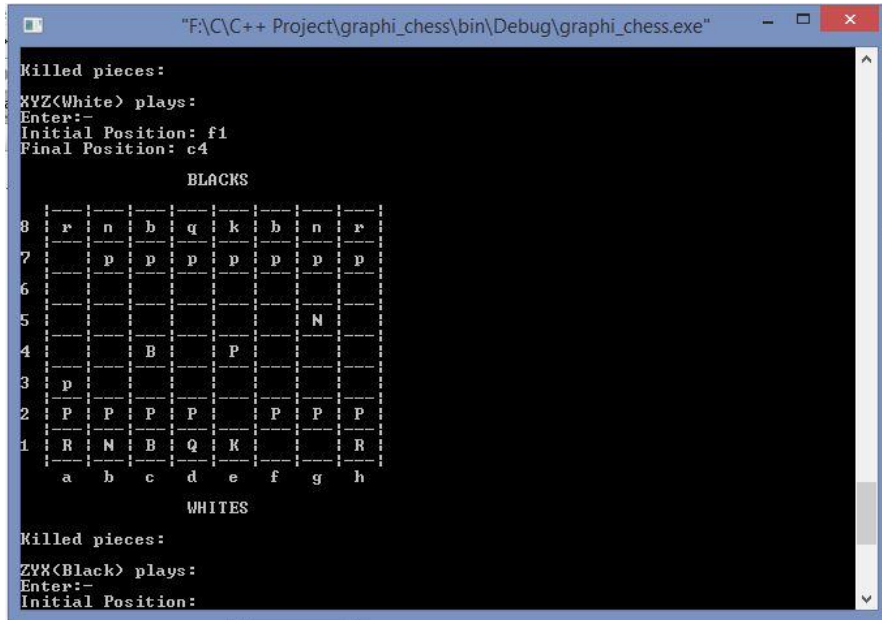
      WHITES
Killed pieces:
ZYX<Black> plays:
Enter:-
Initial Position:
```

```
"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
XYZ<White> plays:
Enter:-
Initial Position: f3
Final Position: g5

      BLACKS
8  r  n  b  q  k  b  n  r
7  _  p  p  p  p  p  p
6  _  _  _  _  _  _  _
5  _  _  _  _  N  _  _
4  p  _  _  P  _  _  _
3  _  _  _  _  _  _  _
2  P  P  P  P  _  P  P
1  R  N  B  Q  K  B  R
   a  b  c  d  e  f  g  h

      WHITES
Killed pieces:
ZYX<Black> plays:
Enter:-
Initial Position:
```

Screenshots: Bishop Move and Pawn Capture



"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"

Killed pieces:

XYZ(White) plays:
Enter:-
Initial Position: f1
Final Position: c4

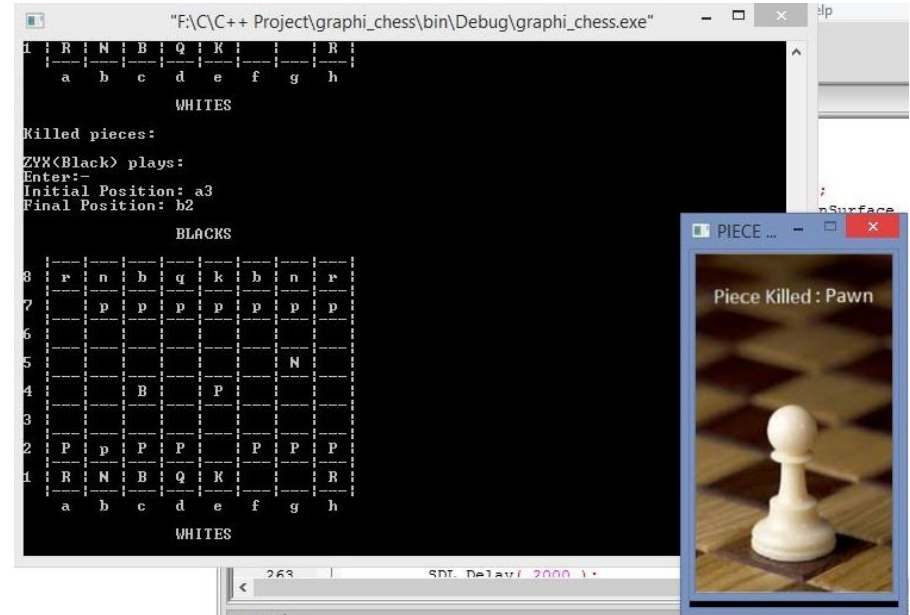
BLACKS

8	r	n	b	q	k	b	n	r
7		p	p	p	p	p	p	p
6								
5							N	
4			B		P			
3	p							
2	P	P	P	P		P	P	P
1	R	N	B	Q	K			R
	a	b	c	d	e	f	g	h

WHITES

Killed pieces:

ZYX(Black) plays:
Enter:-
Initial Position:
Final Position:



"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"

1	R	N	B	Q	K			R
	a	b	c	d	e	f	g	h

WHITES

Killed pieces:

ZYX(Black) plays:
Enter:-
Initial Position: a3
Final Position: b2

BLACKS


8	r	n	b	q	k	b	n	r
7		p	p	p	p	p	p	p
6								
5							N	
4			B		P			
3								
2	P	p	P	P		P	P	P
1	R	N	B	Q	K			R
	a	b	c	d	e	f	g	h

WHITES

263 | SNT Delay: 2000 |

PIECE ...

Piece Killed : Pawn



Screenshot: Queen Move, Pawn promotion and Rook capture

```
"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
XYZ(White) plays:
Enter:-
Initial Position: d1
Final Position: f3
Killed pieces:P,
ZYX(Black) plays:
Enter:-
Initial Position:
Final Position:
```


BLACKS								
8	r	n	b	q	k	b	n	r
7		p	p	p	p	p	p	p
6								
5						N		
4			B		P			
3						Q		
2	P	p	P	P		P	P	P
1	R	N	B		K			R
WHITES								

```
"F:\C\C++ Project\graphi_chess\bin\Debug\graphi_chess.exe"
Killed pieces:P,
ZYX(Black) plays:
Enter:-
Initial Position: b2
Final Position: a1
You can now promote your pawn to any desirable piece.
Enter notation of the desired piece.
q
You just promoted your pawn to a queen
```

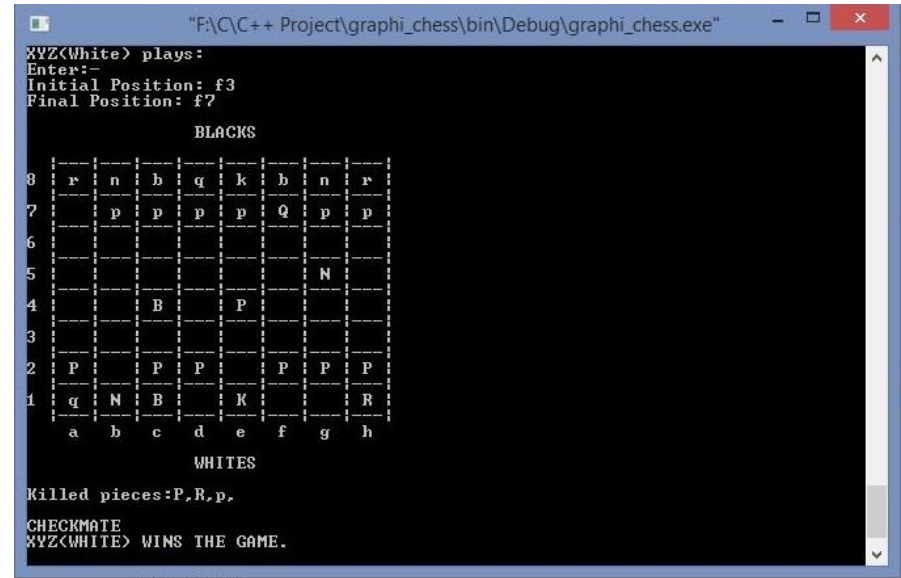
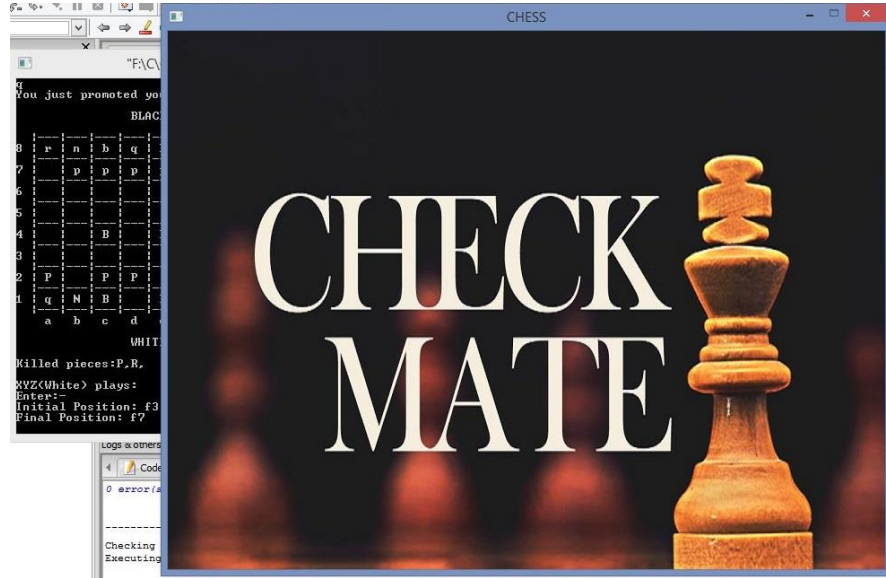
BLACKS								
8	r	n	b	q	k	b	n	r
7		p	p	p	p	p	p	p
6								
5						N		
4			B		P			
3						Q		
2	P	P	P	P	P	P	P	P
1	q	N	B		K			R
WHITES								

PIECE

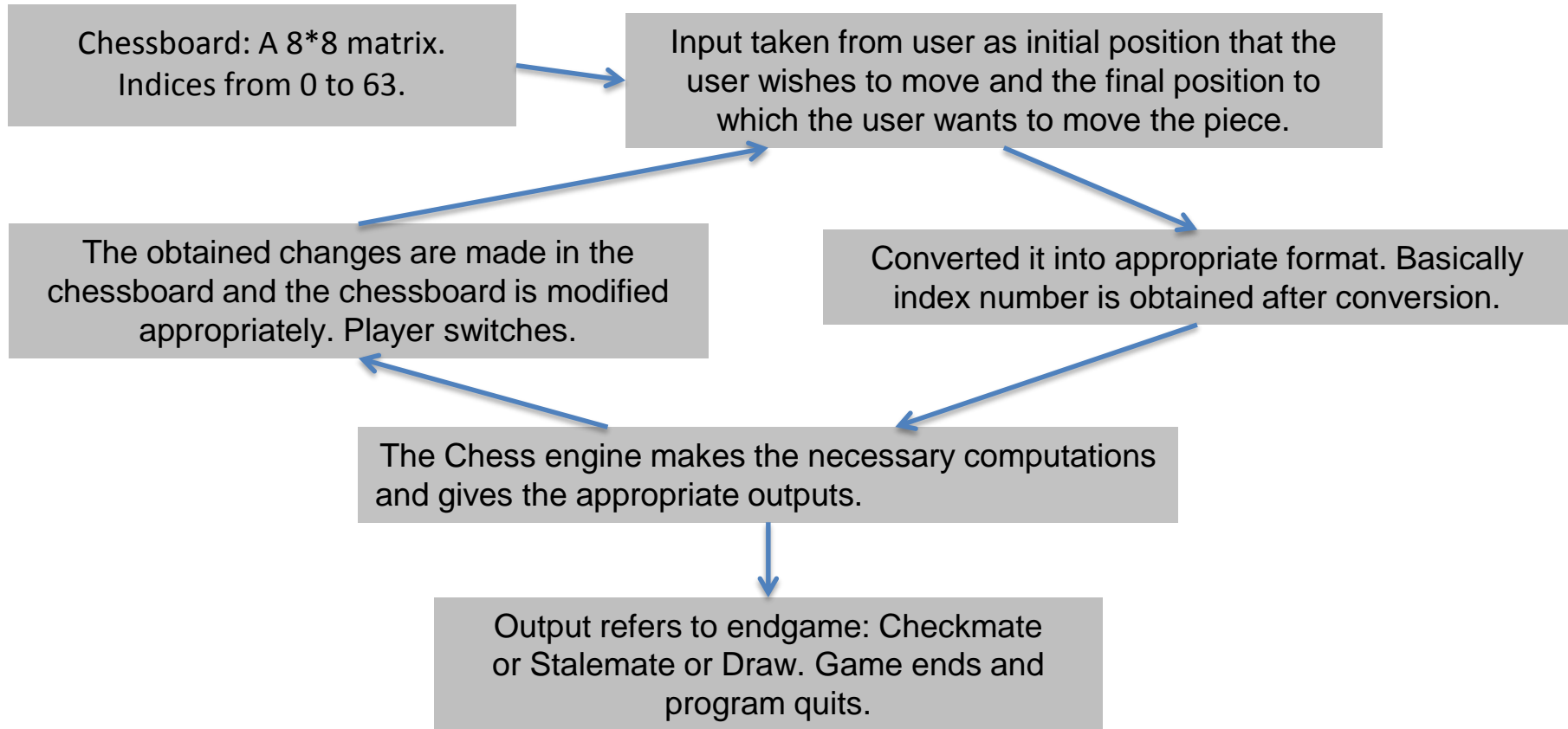
Killed Piece: Rook



Screenshot: Checkmate and Final Chessboard



Algorithm(Game flow)



Algorithm(Engine)

Arguments taken in the form indexes of initial and final positions.

A function computes the move validity. This is the validity disregarding the legality of the moves(i.e. check is not considered here). Another function computes if a capture can be made if the move is valid(as per the above function).

Depending on the outputs of the above functions, the legality of the moves is computed. A move is illegal if it results in the player incurring a check.

Output passed to the board to modify the board appropriately.

It is checked if the output refers to an endgame scenario.

Challenges

- 1) As most of the code is written using pointers, many errors related to pointers such as bad access, memory leaks , etc. occurred.
- 2) Such errors were solved by regular googling of the errors. As many of such errors were faced previously, satisfactory solutions were found on websites like stackoverflow and other forums.
- 3) Implementing graphics was another challenge. As the work on GUI was started pretty late(not a part of the main problem statement), sufficient justice couldn't be given to the GUI.

Future Work

- 1) Improvements can be done in the code itself. A better, shorter and more efficient code can be written.
- 2) A very good GUI can be implemented. Mouse click can be used effectively in place of keyboard inputs.
- 3) A single player AI can be coded. We gave a try to it too, however we fell short of time in understanding the related theorem and then implementing it. The process was left half way by us, however sufficient justice can be given to it in the future.

Thank You