

Assignment 1 - Grading Criteria

1. Parser (10 points)

1.1 Definition of Forsyth Position

The Forsyth language is defined by the following EBNF grammar:

```
Forsyth ::= { Row ; } Row .  
Row ::= { WPiece | BPiece | Digit }  
WPiece ::= R | N | B | Q | K | P  
BPiece ::= r | n | b | q | k | p  
Digit ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
```

A whitespace character is defined as any single character representing horizontal or vertical space in typography, and shall be ignored during the parsing process. According to the grammar, a row is a sequence of white pieces (WPiece), black pieces (BPiece), and digits. Sequences of digits are not interpreted as numbers. Therefore, the string "12" is interpreted as one empty square followed by two empty squares. A Forsyth configuration consists of a sequence of rows delimited by semicolons. As a special case, the last row terminates with a point.

In order to be valid, a Forsyth position file has to fulfill the following conditions:

1. Exactly 8 rows shall be specified.
2. Each row shall specify the state of exactly 8 squares.
3. Each player shall have exactly one king.

A row not explicitly stating the state of exactly 8 squares is incorrect. Omitting to mention trailing empty squares is therefore also an error. For instance, the sequence "pp;" is not a correct description for a row, and shall be rewritten as "pp6;".

Incorrect input shall be reported to the user. The message shall include the cause of the problem. Under no circumstance the application is allowed to exit abruptly, throwing an exception. In particular, the application shall not crash if the input file contains bad characters.

1.2 Definition of the Algebraic Notation

```
Move ::= [ Piece ] Position ( - | x ) Position [ + | # | ++ ]
```

```
Piece ::= R | N | B | Q | K
```

```
Position ::= ( a | b | c | d | e | f | g | h ) ( 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 )
```

A move in the algebraic notation consists of the type of the piece that is moved, the original position, either the connector – or x, a destination, and the checkmate status. The type of the piece is omitted for pawns only. The connector x is used to notify that a piece has been captured.

Otherwise, the connector – shall be used. Finally, Either # or ++ shall be set in checkmate positions. The indication that the king is under attack but has the possibility escape is written using a +. Omitting the checkmate status is considered as an error. Finally, castling moves are reported in the following way:

Move ::= O – O [+ | # | ++] | O – O – O [+ | # | ++]

Castling to the King's side is written “O-O” while castling to the queen's side is noted “O-O-O”.

1.3 Accept command line parameters (2 points)

The program has to be executable from the command line in the following manner:

```
Chess.exe <inputFile> <outputFile>
```

Command line parameters:

<inputFile> The input file which contains a position in Forsyth notation.
<outputFile> The output file that stores the generated move for the white player.

Example 1: **Chess.exe** **position.txt** **white_action.txt**

Example 2: **Chess.exe** **myInput.txt** **myOutput.txt**

Example 3: **Chess.exe** **in.txt** **out.txt**

Note: The program must not read the file “position.txt” by default and must not write to a file named “white_action.txt” by default. Instead it has to read two command line parameters which define the input file and the output file. The program has to read the position from the specified input file and write the result to the specified output file. Finally, the application shall not crash if the commandline is incorrect, or if the files cannot be open. A meaningful message must be outputted instead.

1.4 Accept Forsyth position file (5 points)

The parser has to accept a correct Forsyth position file as defined in 1.1. Further it has to check if the position defined in the file is correct. If the position defined in the input file is not correct, the program has to output a message to the user.

Note: The program must not crash upon an incorrect input file.
The program must not throw an exception on an incorrect input file. This is a misuse of exception handling.
In such cases the program has to output an error message to the user.

1.5 Output file format (3 points)

The output file has to contain one legal move of the white player given a valid and correct position file. The move shall be represented in the algebraic notation. Any other format of output is not accepted.

Note: Valid and correct means that the syntax of the Forsyth file conforms to the given EBNF, and the positions for each piece listed in the input file are all potential positions of the pieces.

2. Class diagram (10 points)

The class diagram can be drawn with an UML drawing tool or the class diagram generated with Visual Studio 2005 can be used. The class diagram has to contain all properties, methods and members of a class.

Note: The class diagram must not be drawn by hand.

3. Search algorithm (10 Points)

Do not consider the situation that the game is drawn.

The program has to correctly implement all chess rules except for En Passant and Castling, because they depend on the move history. For a correct Forsyth file, the program should generate a legal move.

Note: Failing to protect the king in a non checkmate situation or exposing the king to the adversary are illegal moves.

4. Reduction of points

Unhandled exceptions are considered as crashes and points will be deducted in such cases.