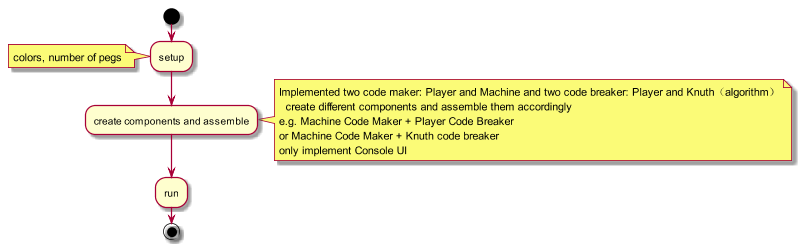
**Zhou Rui (Jon)**

**(**+65) 85187866 **|** **wirelesser@hotmail.com**

# Introduction

Refer to https://en.wikipedia.org/wiki/Mastermind\_(board\_game),The mastermind program is mainly consist of 4 components: UI, code breaker, code maker and main. UI is implemented as console UI; for code breaker I have implemented knuth algorithm and player(user);for code maker I also implemented a machine code maker and a player code maker as well. Since these components have abstract class interface, I can easily create these components and assemble them accordingly, and play the game in different modes, e.g. knuth code breaker vs machine code maker. There is just little code difference in main routine:



# Build

Tested with GCC 4.8.4 @ Linux ubuntu 3.16.0-30-generic

Set strict compile flags like –Werror, -Wconversion, and turn on -std=c++0x in CMakeLists.txt

For other compilers, should support these C++11 features:

1. shared\_ptr
2. lambda
3. std::function<>
4. auto
5. explicitly defaulted and deleted special member functions
6. noexcept, override

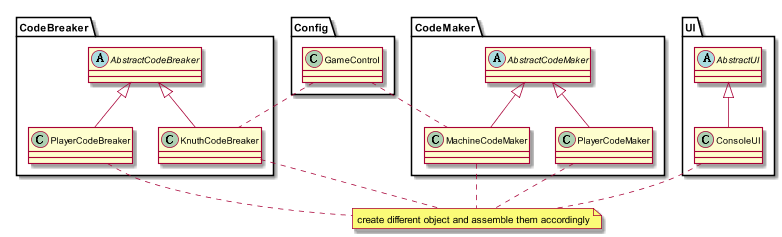
Build is straightaway,

\*\*Steps\*\*

1. create build folder;”mkdir bld;cd bld”
2. cmake PATH/TO/CMakeLists.txt (e.g. cmake -DCMAKE\_BUILD\_TYPE=Debug ../;)
3. make

now everything is under “bld/bin/” folder.

# Basic Class Diagram



# Unit test and integration test

run “bin/unit\_test” for unit test.

Since I don’t include any 3rd unit test framework(e.g. GTest), I wrote some very basic unit test helper functions/class.

Output will be like this(a failed test case is intend to be there):

ruizhou@ubuntu:~/greenwave/bld/bin$ ./unit\_test

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

KnuthCodeBreaker Unit Test

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

test constructor passed

test pattern combination generation passed

test guess passed

test feedback failed

KnuthCodeBreakerTest passed 3 failed 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MachineCodeMaker Unit Test

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

test constructor passed

test secret code generation passed

test make guess passed

MachineCodeMakerTest passed 3 failed 0

I don’t include any integration test/mock either. But we can look at the file “code\_breaker\_knuth.cpp” and “code\_maker\_machine.cpp”, and the executable binary file at “bin/knuthcb\_machinecm”, it is game played between machine(knuth is code breaker and machine is code maker).it can be automatically running until knuth guess the code without manual intervention.

# FileList

|  |  |
| --- | --- |
| Src/ |  |
| abstract\_ui.h | abstract base class of UI |
| console\_ui.h | implementation of console UI |
| abstract\_codebreaker.h | abstract base class of code breaker |
| code\_breaker\_knuth(h/cpp) | implementation of Knuth code breaker algorithm |
| code\_breaker\_player.h | implementation of player code breaker |
| abstract\_codemaker.h | abstract base class of Code Maker |
| code\_maker\_machine(h/cpp) | implementation of machine code maker |
| code\_maker\_player(h/cpp) | implementation of player code maker |
| match\_algo.h | class implementation to count number of color matched and number of color and position matched |
| game\_control(h/cpp) | game control data |
| main\_knuthcb\_machinecm.cpp | Game: Knuth code breaker vs machine code maker |
| main\_knuthcb\_playercm.cpp | Game: knuth code breaker vs player code maker |
| main\_playercb\_machinecm.cpp | Game: player code breaker vs machine code maker |
| main\_playercb\_playercm.cpp | Game: player code breaker vs player code maker |

|  |  |
| --- | --- |
| Test/ |  |
| knuth\_codebreaker\_test(h/cpp) | Knuth code breaker unit test |
| machine\_codemaker\_test(h/cpp) | Machine code maker unit test |
| main\_unittest.cpp | Unit test main routine |

# Play game

## See how does knuth code breaker help to play game

Here is a funny test. We can start two games at the same time, one is playercb\_machinecm (player code breaker vs machine code maker), another one is knuthcb\_playercm (knuth code breaker vs player code maker), then copy the knuth’s guess to help to play the former game.

\*\*player code breaker vs machine code maker\*\*

ruizhou@ubuntu:~/greenwave/bld/bin$ ./playercb\_machinecm

Configuring the game...

please input a string each char of the string represent one color, e.g. 'RGBP'...RGBP

please input the number of pegs...4

Code Breaker,input your guess...RRRR

Code Breaker guess RRRR

Code Maker feedback(color match,color and position match): 1,1

Code Breaker,input your guess...RGGG

Code Breaker guess RGGG

Code Maker feedback(color match,color and position match): 2,0

Code Breaker,input your guess...GRBB

Code Breaker guess GRBB

Code Maker feedback(color match,color and position match): 4,2

Code Breaker,input your guess...GBRB

Code Breaker guess GBRB

solved in 4 steps!

\*\*knuth code breaker vs player code maker\*\*

ruizhou@ubuntu:~/greenwave/bld/bin$ ./knuthcb\_playercm

Configuring the game...

please input a string each char of the string represent one color, e.g. 'RGBP'...RGBP

please input the number of pegs...4

Code Breaker,input your guess...

Code Breaker guess RRRR

Code Maker,give feedback on RRRR(color\_match color\_pos\_match)...1 1

Code Maker feedback(color match,color and position match): 1,1

Code Breaker,input your guess...

Code Breaker guess RGGG

Code Maker,give feedback on RGGG(color\_match color\_pos\_match)...2 0

Code Maker feedback(color match,color and position match): 2,0

Code Breaker,input your guess...

Code Breaker guess GRBB

Code Maker,give feedback on GRBB(color\_match color\_pos\_match)...4 2

Code Maker feedback(color match,color and position match): 4,2

Code Breaker,input your guess...

Code Breaker guess GBRB

Code Maker,give feedback on GBRB(color\_match color\_pos\_match)...