COMP 345 Assignment 1
Grading Schema

Grading schema for COMP345, Fall2018 section D given by Dr. Joey Paquet

Non-implementation Part (8 points)

Knowledge/correctness of game rules (2 points)

During the presentation/demo if you make the marker aware that you don't understand the game rule properly or if the code implements incorrect rules (according to the official game rules), remove marks.

Modularity/simplicity/clarity of solution (2 points)

Data structures should be appropriate, simple, and clear. If have difficulties explaining it, it is unclear.

Proper use of language/libraries (2 points)

Should correctly use .h and .cpp files. Properly use the technique to avoid multiple inclusion. etc.

Code readability (2 points)

Improper naming, messy code layout, commented-out code, etc. will result in mark deductions.

Implementation Part (12 points)

Map (3 points)

1. using any kind of graph traversal algorithm to prove the graph you generate (including sub graph) is a connected graph (it means you should be able to traversal any continent or the whole world) [2 points];
2. each country should belong to one and ONLY one continent [1 point];

MapLoader (2.5 points)

1. can successfully parse any map file from the "Conquest" game web site, you are NOT allow to hard code the graph or using your map file format (during the

demo marker may ask you a load a pre-defined map for testing, you code should be able to change the target map file easily) [1.5 points];
2. can report the map file is invalid if it is [1 point];

Dice (2.5 points)

1. the dice rolling facility should have a member function that receive an integer from 1 to 3 to decide how many dice are being rolled [1 point];
2. you should keep tracing the dice rolling percentage [1 point];
3. the return value of each dice should be in the range 1 ~ 6 [0.5 point];

Player (2 point = 4 * 0.5)

1. each player owns a collection of countries;
2. each player owns a collection of Risk cards;
3. each player has their own dice facility;
4. each player has: reinforce(), attack(), fortify() interface (empty function is good for this submission);

Card deck / hand (2 points = 0.5 point * 4)

1. the deck card number should be equal to the number of countries in the map;
2. each card has a type from: infantry, artillery, cavalry;
3. has a draw() method can return the card you draw from the deck;
4. exchange() method in hand can exchange armies using the Risk card;