An intelligent Agent for the board game Risk

# Introduction

Risk is a strategic turn-based game for two to six players. The standard version is played on a map of the Earth which is divided into forty two territories. These territories are grouped into six continents. (Asia, North America, Europe, Africa, Australia, South America).

The goal of the game depends on a player’s given mission. The available missions include conquering specific continents, conquering a given number of countries or wiping out a specific enemy. There is also a variant of the game called “Wolrd Domination” which assigns the same mission to all participating players: conquering the whole world by defeating all other players.

Players use armies which have a distinct color for each player to conquer and defend countries. Dependent on the strategy the player plan the placement and the use of this armies Depending on his preferred strategy the player distributes his armies between his countries. In general this means either distributing armies equally to all countries or just putting all of them in one country to use them as an attack army. Winning depends partly on the strategy used and luck at throwing the dices.

# Rules

## Preparations

First the game principle has to be decided. There are two alternatives.

1. “World domination”
2. Cards with different missions (conquer special continents, conquer specific countries, defeat a specific player)

At the beginning of the game so called Risk Cards are mixed up handed out to all players clockwise. It is possible that players receive unequal amounts of cards. Risk Cards display a country and an symbol (either a soldier, a cavalryman or a canon). The players go through all their cards and place one army on each of the countries displayed on the cards. There needs to be at least one army on each country at all times.

After all countries have been filled with one army the Risk Cards are recollected and mixep up together with two joker cards (cards which contain all three symbols).

With the second alternative each player would also be given one mission card.

With all preparations done the normal game play begins.

## Game Play

Risk is a turn based game beginning with the player left to the one who received the last Risk Card. Each turn is divided into 3 phases:

1. New Armies
2. Conquering Other Countries
3. Relocate Armies

### 1. Phase: New Armies

The player receives new armies which he can place on his countries. The amount of armies depends on how many countries the player controls. The number of countries is divided by 3 with the result defining the number of armies (decimals are ignored). E.g. a player holds 14 countries. The equation would be 14 / 3 = 4.67 and therefore result in the player receiving 4 armies. Even though a player might have too few countries he always receives at least 3 armies.

|  |  |
| --- | --- |
| Continent | Bonus Armies |
| Africa | 3 |
| Asia | 7 |
| Australia | 2 |
| Europe | 5 |
| North America | 5 |
| South America | 2 |

In addition to this armies a player can receive bonus armies for holding whole continents. The number of bonus armies depends on the continent. The table on the right shows the bonus armies per continent.

If a player has at least 3 Risk Cards he may be able to exchange them for additional armies. Therefore he needs either 3 cards with the same symbol (soldier, cavalryman or canon) or 3 cards with each showing a different symbol. A joker card is equivalent to any of the 3 symbols. The number of armies received for exchanging cards raises each time a player does so. The table below shows how many armies are received based on the number of exchanges. After the 6th exchange each new exchange yields 5 more armies than the previous one.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number of Exchange | 1st | 2nd | 3rd | 4th | 5th | 6th |
| Armies | 4 | 6 | 8 | 10 | 12 | 15 |

The player has to place all received armies directly and can not store them for later turns. He can decide himself on which of his countries he places his armies. He could also place all armies on only one of his countries.

### 2. Phase: Conquering Other Countries

The player can use each of his countries with more than 1 army inside to attack other countries that are directly adjacent to the country from which he attacks or have a connection by sea (lines on the map).

During the attack victory and defeat is decided by dice. The attacker can use up to 3 dices with each dice representing one army. Since one army needs to be left behind for defense purposes this one can not be used during the attack e.g. if the player has 3 armies in a country which he uses for the attack he will only be able to use 2 of them in the actual attack resulting in using only 2 dices. To use all 3 dices he needs at least 4 armies in the country. Having more than 4 armies in the country only results in more throws with 3 dices but not in using more than 3 dices.

The defending player can use all of the armies inside the attacked country but with a maximum of 2 dices.

After the dices have been thrown their values are compared dice for dice. To win the attacker needs to have a higher values. Equal values with the defenders dice will result in victory of the defender.

E.g.: The attacker throws 3 dices resulting in a 5, a 4 and a 2. The defender throws two dices having a 5 and a 3. This would result in attacker and defender loosing one army each because 5 is equal to 5 (attacker looses one army) and 4 is higher than 3 (defender looses one army).

Throwing dices continues until all armies in the attacked country or all but one army in the attacking country have been eliminated. The attacking player can also decide to stop attacking if it suits him.

If the attacker succeeds in conquering the country he has to move all armies who took part in the last throwing of dices to the conquered country (which means depending on used dices either 1, 2 or 3 armies). He is allowed to move additional armies to this country from the country which he used for the attack as long as at least one army is still in the origin country. The player can now run further attacks from the conquered country or any other country with at least 2 armies.

If the player defeats the last army of another player the defeated player leaves the game and the player who defeated him gets all his Risk Cards with the option to directly exchange cards (his own and the newly gained ones) for armies. These armies need to be placed directly.

### 3.Phase: Reposition Armies

When a player is finished with attacking other countries he can reposition remaining armies (while leaving at least one army in each of his countries). All armies that have not been used during attacks can be repositioned. Repositioning is only possible between neighbored countries of the player.

When the player is done repositioning armies the turn of the player to his left begins his turn with the first phase.

## End of Game

Depending on the chosen alternative the end of game varies. With “World Domination” the game ends as soon as one player has conquered the all countries of the world. This player wins the game.

In the variant with missions the game ends when one player fulfills his mission. He does not necessarily need to do so himself e.g. when the player with green armies defeats the last country of the red player and the blue player has the mission to defeat all red armies, the blue player would win.

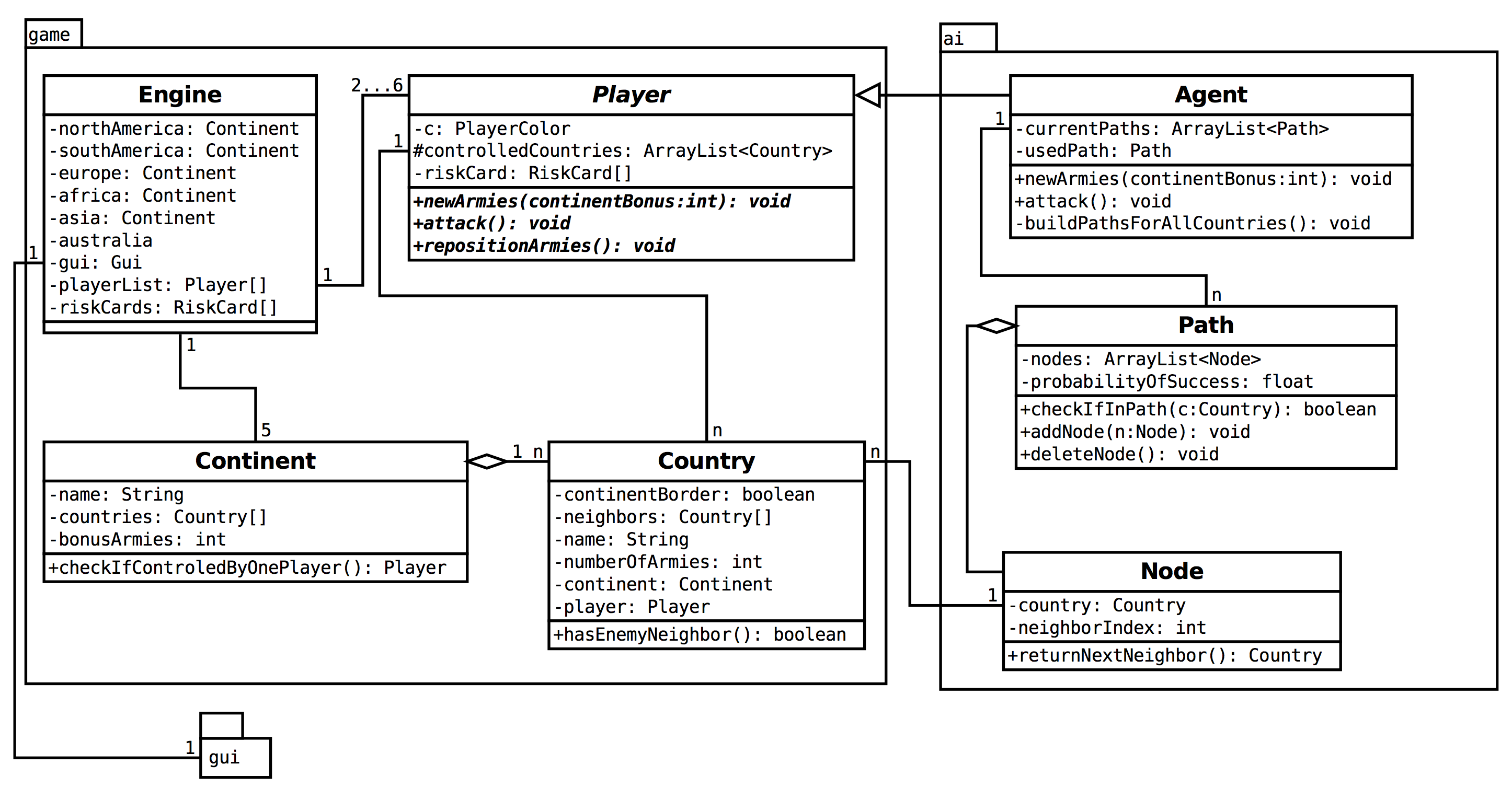
If two players complete their mission at the same time the winner will be the player who actually completed his mission himself as the following example shows:

The blue player has the mission to conquer Asia and Africa. He currently controls all of Asia and all but one country of Africa. The last country of Africa is controlled by the red player. This is also the last country the red player controls. The green player has the mission to defeat all red armies. When the blue player defeats this last country of the red player both the green and the blue player’s mission is completed. Since the blue player actually made the last move he wins the game.

# Implementation

To be able to develop an intelligent solving agent for the game it is necessary to implement the game first with an object model that allows an agent to determine its attack strategy.

The used object model is explained here.



The class *Engine* represents the actual game logic. It has all *Continent* objects which themselves have the corresponding *Country* objects. Each *Country* object also knows all *Country* objects that are neighbors of it on the actual map. For the graphical representation all *Country* objects also have their own *ImageLabel* object which is used to display the corresponding image file in the correct color and the number of armies placed inside the country.

The *Engine* also knows all *Player* objects that represent the players with each of them knowing their *PlayerColor* and the *Country* objects controlled by them. The *Agent* extends the *Player* objects and implements the algorithm described later in this document. By adding other subclasses it would also be possible to add human controled *Player* objects to the game. Since this feature is not of any relevance to this project it will not be implemented at this stage. Each *Player* also has up to five *RiskCard* objects while the *Engine* has all other *RiskCard* objects.

For general parts of the GUI like displaying the game map or configuration interface the class *Gui* is used. The *Gui* is also controlled by the *Engine*.

Once the game is started and the number of different *Player* objects is set by the user the *Engine* will check all with continents whether they are controlled by a single player in order to allow the use of continent bonus armies. Afterwards it will invoke the action methods (*newArmies(), attack(), repositionArmies()*) of one *Player*. After each turn (meaning all 3 action methods have been called on one *Player*) the *Engine* checks if the end condition is met which is that one *Player* controls all 42 *Country* objects. If true the game is finished. If false the *Engine* will switch to the next *Player* and repeat checking for complete continent control and invoking the action methods on the new player.

**int** currentPlayerIndex = 0;

**while** (!gameFinished()) {

Player currentPlayer = playerList[currentPlayerIndex];

currentPlayer.newArmies(

checkContinentsForSingleOwner(currentPlayer));

currentPlayer.attack();

currentPlayer.repositionArmies();

currentPlayerIndex++;

**if** (currentPlayerIndex >= playerList.length) {

currentPlayerIndex = 0;

}

}

System.*out*.println("End of Game.");

The action methods (which have to be implemented by subclasses of *Player* since *Player* itself defines them as abstract methods) define how the given *Player* object handles the 3 phases of its turn.

In the *newArmies()* method it needs to decide whether it wants to exchange its RiskCards (if possible) and on which countries to put received armies. The amount of armies resulting from controlled countries and if applicable continents is calculated automatically.

In the *attack()* method the *Player* subclass needs to define which attack actions shall be taken. This method also needs to implement the case that an enemy player is completely destroyed and the current player needs to exchange surplus RiskCards.

In the *repositionArmies()* method has to be an implementation for how the *Player* repositions remaining armies;

# Developing an Intelligent Agent for Risk

Since using mission cards adds enormously more complexity to the rules of an intelligent agent this project will focus on the alternative “World Domination”.

At first it needs to be decided which kind of strategy the agent should use. There are different approaches to playing Risk.

One approach would be to distribute all received armies equally to all countries. The result would be that all countries contain the same number of armies with some having one less because there were not sufficient armies available to distribute exactly the same amount on all countries. During the second phase (conquering other countries) an agent using this strategy would always conquer only one country which contains preferably as few armies as possible. In the third phase (repositioning of armies) it would try to balance the amount of armies in all countries as far as possible.

Another approach is to put all received armies in one country as an attack army. A modification of this strategy would be to also defend the border countries of continents before putting all armies inside one attack country. This results in better defended continent borders (and therefore probably more bonus armies in the next round if the player controls the whole continent) but a weaker attack army.

During the second phase the agent chooses a path of countries to conquer. This decision is based on an algorithm described below.

During the reposition phase the agent would do nothing since he already defended continent borders in the first step and fights with his attack army until it is completely defeated or it has won the game.

The second strategy shall be realized during this project in an example program. The agent will also use the modification with continent border defense by distributing 50 percent of his received armies to continent border countries while using the remaining 50 percent as an attack army.

## Path Finding Algorithm

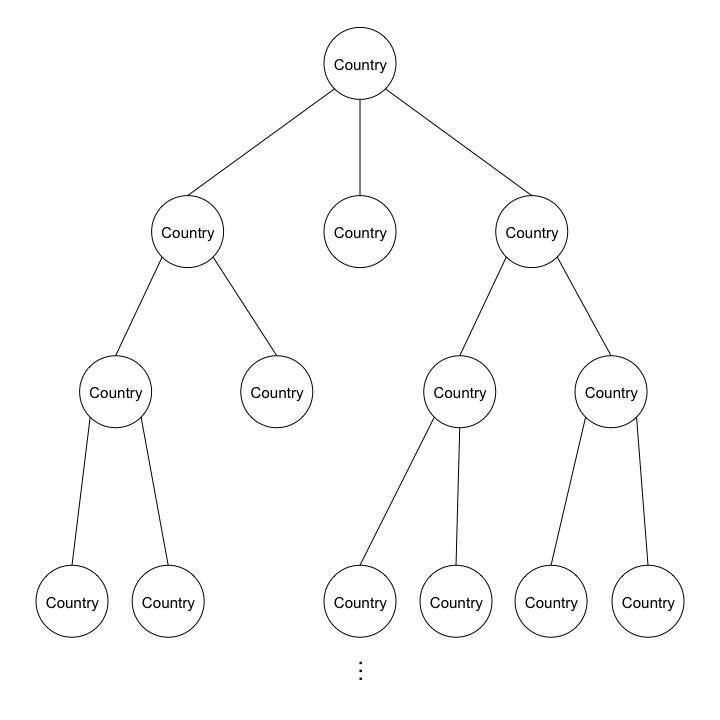
As mentioned before an intelligent agent needs to consider several factors in the decision of an attack path. The list below shows the most important ones.

* Theoretical path length (number of countries that can be conquered in a chain)
* Number of enemy armies on the path
* Number of own armies
* Conquering of whole continents on the path

Since the last factor is difficult to realize programmatically only the first three shall be considered during implementation.

Since an intelligent agent does not really have a “view” of the map it is necessary to use a data structure that helps the agent find a path. In this case the most useful solution is a tree structure to represent all possible paths. Note here that these trees are never saved as a data structure. Instead the agent builds paths by moving through the country objects like they where in a tree structure.

shows an example view of a tree how it is used here. The “root” country is the country from which the player is going to attack. On the first leave level are all direct neighbor countries which are controlled by another player. On the second leave level are all neighbor countries of the ones on the first leave level. These are also controlled by enemy players. This principle continues throughout the whole tree. Countries controlled by the player himself are never added to the tree. All routes from the root country to the leaves in the tree (with no other leaves following them) are each considered a separate path which means that there are as many paths as there are leaves with no successors. A country may appear multiple times in the tree but it never appears more than once in a single path.

 Figure : Path Tree

shows the beginning of the path tree for the country New Guinea. It can be seen that some countries appear multiple times in the tree but never in the same path. Note that in this example all countries except New Guinea do not belong to the player who is attacking.

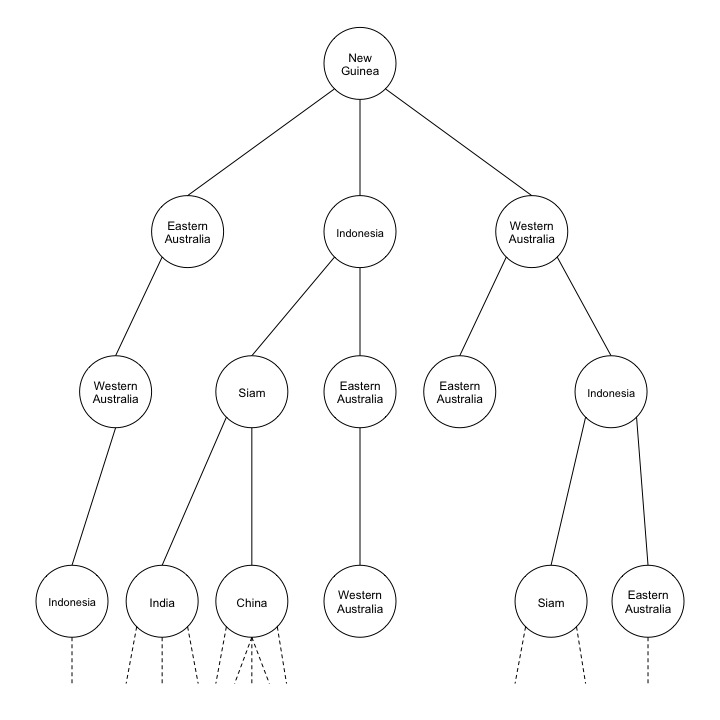


Figure : Example Tree for New Guinea

At the beginning of its turn the agent builds up all possible paths for each country controlled by it that has at least two armies placed in it and has at least one neighbor countries controlled by another player. The path length limit depends on the number of armies available for the attack e.g. with 10 attack armies the agent can conquer a maximum of 10 countries (assuming that no army is lost in battle). Paths will only be build up to this limit to avoid unnecessary path calculations.

currentPaths.clear();

**for** (Country c : **this**.controledCountries) {

**if** (c == **null**) {

**continue**;

}

**if** (c.hasEnemyNeighbor() && c.getNumberOfArmies() > 1) {

Path p = **new** Path(c);

Country t = p.getCurrentNode().returnNextNeighbor();

**while** (!(p.getCurrentPosition() == 0 && t == **null**)) {

**while** (t != **null**) {

**if** (p.armiesLeft()) {

**if**(!p.checkIfInPath(t)) {

p.addNode(**new** Node(t));

}

t = p.getCurrentNode().returnNextNeighbor();

} **else** {

t = **null**;

}

}

currentPaths.add(p.clone());

**while** (t == **null**) {

**if** (p.getCurrentPosition() == 0) {

**break**;

}

p.deleteNode();

t = p.getCurrentNode().returnNextNeighbor();

}

}

}

}

After all trees have been built the agent compares the available paths and decides which one it will take based on the possibility of reaching the last country in the path with his attack. Therefore it considers the number of its own armies as well as the number of enemy armies along the path.

Once it has chosen one path it will begin with the attack along the path until all his armies are used. It will always move all armies towards the next country. This will result in the agent only stopping if the attack army is destroyed to only one army left or it reached the end of the path.