SOEN 6441
Advanced Programming Practices
SECTION DD
WINTER 2010

Project Rules

Troy Taillefer 3499502
Justin Stefani 4305787
Pavel Lepin 9711082
Iouri Goussev 5484750

Table of Contents

Table of Figures

General Rules

General:

When the game starts, all states and cities must have an owner (player).

There is only one type of resources: Iron. If a state or a city has iron, it also has an iron mine. (In other words, no needs to build the iron mine). Each mine produces 100 kg of iron every turn.

There are two type of state: state and city. In cities production factories can be built.

Each state has attributes:

Basic attack battle score

= (infantry.Count*1+ cavalry.Count*3 + artillery.Count*5)

Basic defense battle score

= (infantry.Count*2+ cavalry.Count*2 + artillery.Count*6)

Risk rate

= the rate of the sum of enemy states **Basic attack battle scores** and its own **Basic defense battle score.** Zero means there are no connected enemy state. Value greater that 1 means that the state probably will be attacked. States with the rate less then 1 will never be attacked.

Extra battle scores

= the difference between its own **Basic defense battle score** and the sum of enemy's states **Basic attack battle scores**

Production Rules

Factory construction. Each city can build one building in a turn. Building costs:

- barracks: nothing
- stables: 200 kg of iron (can be build after barracks)
- artillery factory: 500 kg of iron (can be build after stables)

2. The Player's Production Score is calculated. The Formula could be: **AConstant + States.Count + Cities.Count*2 + Contintnts.Count*5**

   Then scores are divided equally between cities (because only cities can produce armies). After that cities try to produce the most powerful units as much as it can, then less powerful ones. Unit price:

   - infantry: 1 Production Unit
   - cavalry: 2 Production Unit
   - artillery: 3 Production Unit + 100 kg of iron

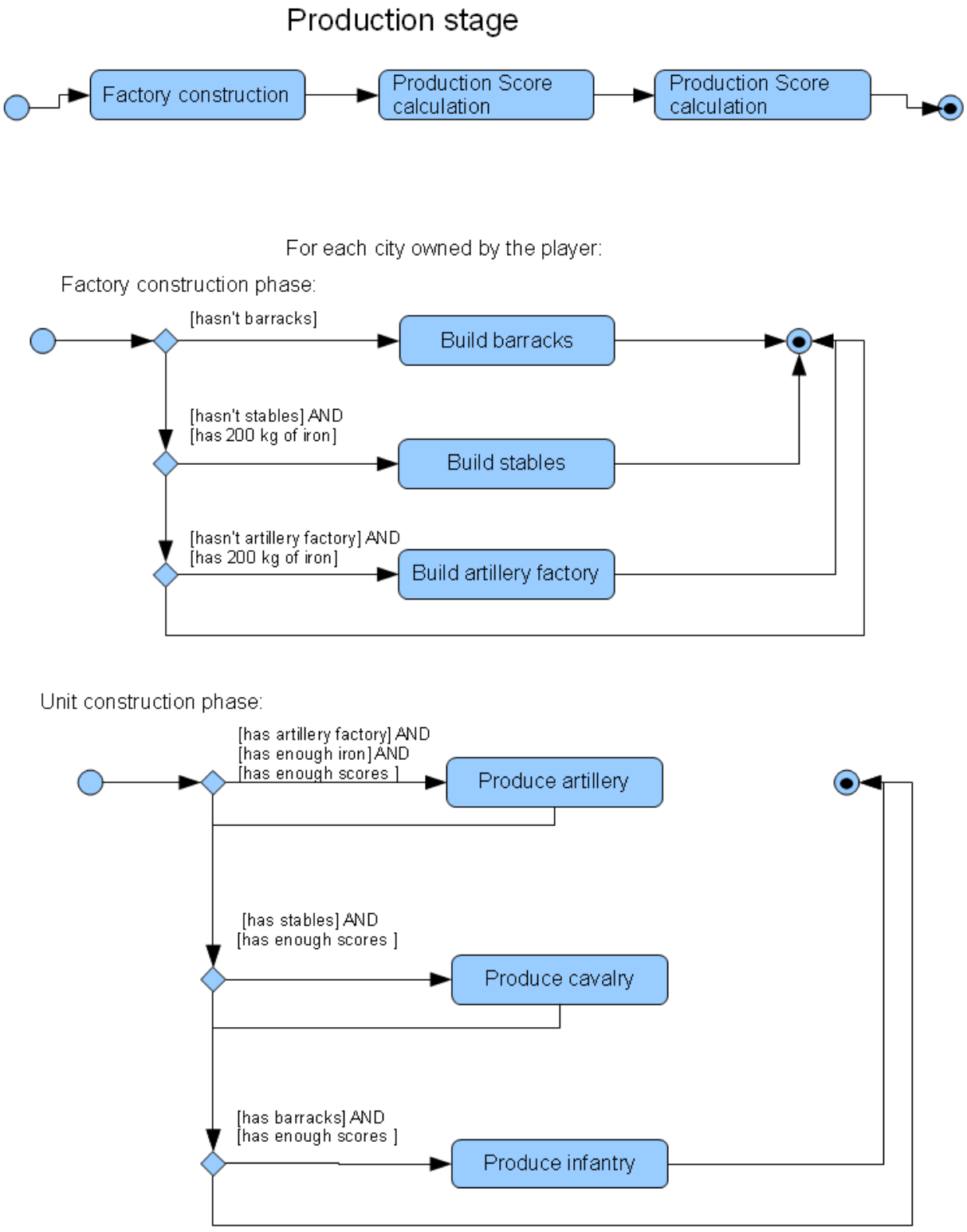

Illustration 1: Production State Diagrams

Battle Rules

Every state that can be attacked (all connected enemy states) is assigned a **willBeAttacked** rate. The rate formula:

[hasIron coefficient] + [isCity coefficient] + **risk rate.**

hasIron and ***isCity*** coefficients should be in range [0.1..0.5]

All enemy states with **willBeAttacked** rate greater then 1 is subject to attack, starting from the state with the largest **willBeAttacked** rate. Attacking state has to have **Basic Attacker battle score >= enemy state's Basic Defender battle score.**

Battle:

1. When one state attacks another one, one infantry unit stays home (does not participate in the battle).
2. Battle scores are calculated for the attacker and defender for every battle turn:

   Attacker battle score = (infantry.Count*1* *luck* + cavalry.Count*3* *luck* + artillery.Count*5* *luck*)

   Defender battle score = (infantry.Count*2* *luck* + cavalry.Count*2* *luck* + artillery.Count*6* *luck*)

   Where "*luck*" is a random number (i.e. from 0.1 to 1). The *luck* is recalculated for every operation.
3. Then damage is calculated. It starts from most powerful to less powerful (if battle scores are not enough to destroy more powerful one). The damage formula could be:

   Attacker:

   Artillery_killed = (Defender battle score) % 5;

   If (Artillery_killed < artillery.Count) Artillery_killed = artillery.Count;

   artillery.Count = artillery.Count - Artillery_killed;

   Defender battle score = Defender battle score – Artillery_killed * 5;

   cavalry_killed = (Defender battle score) % 3;

   If (cavalry_killed < artillery.Count) cavalry_killed = artillery.Count;

   cavalry.Count = cavalry.Count - cavalry_killed;

   Defender battle score = Defender battle score - cavalry_killed * 3;

infantry.Count = infantry.Count - (Defender battle score) % 1;

Defender:

Artillery_killed = (Defender battle score) % 6;

If (Artillery_killed < artillery.Count) Artillery_killed = artillery.Count;

artillery.Count = artillery.Count - Artillery_killed;

Attacker battle score = Attacker battle score - Artillery_killed * 6;

cavalry_killed = (Defender battle score) % 4;

If (cavalry_killed < artillery.Count) cavalry_killed = artillery.Count;

cavalry.Count = cavalry.Count - cavalry_killed;

Attacker battle score = Attacker battle score - cavalry_killed * 4;

infantry.Count = infantry.Count - (Attacker battle score) % 2;

4. The attacker calculates the sense to continue the attack. The formula:

Do attack again = (**Basic Attacker battle score >= Basic Defender battle score).** If (Attack again) then repeat attack from step 2.

5. Battle result:

Attacker lost (or has decided to stop the attack): all left units go back to the initial state.

Attacker won: If more then 50% of neighbors of the initial state are own by same player, 80% of attacker forces move to the new state, otherwise only 49% go to the new state.

Example:

Let's say State A attacks State B. State A is surrounded by 2 friendly and 2 enemy states.

State A has 3 artillery, 3 cavalry and 5 infantry units. (Attacker has 4 infantry units – one stays in the State A)

State B has 3 artillery, 3 cavalry and 2 infantry units.

The Battle:

Step 2:

Attacker battle score = (4*1* *0.8* + 3*3**0.5* + 3*5**0.9*) = 0.8+4.5+6=21.2

Defender battle score = (2*1* *0.7* + 3*3* *0.4* + 2*5**0.6*) = 1.4+3.6+6=11

Step 3:

Attacker:

Artillery_killed = 11 % 5 = 2;

artillery.Count = 3 – 2 **=** 1;

Defender battle score = 11 - (2 * 5) = 1;

cavalry_killed = 1 % 4 = 0;

cavalry.Count = 3 – 0 = 3;

Defender battle score = 1 - (0 * 3) = 1;

infantry.Count = 4 – 1%1 = 3;

Defender:

Artillery_killed = 21.2 % 5 = 4;

Artillery_killed = 3; // less then *artillery.Count*

artillery.Count = 3 – 3 **= 0**;

Attacker battle score = 21.2 - (3 * 5) = 6.2;

cavalry_killed = 6.2 % 3 = 2;

cavalry.Count = 3 – 2 = 1;

Attacker battle score = 6.2 - (2 * 3) = 0.2;

infantry.Count = 2 – (0.2 % 2) = 2;

Now

Attacker has 1 artillery, 3 cavalry and 3 infantry units.

Defender has 0 artillery, 1 cavalry and 2 infantry units.

Step 4:

Attacker battle score = (3*1 + 3*3 + 1*5) = 3+9+5=17

Defender battle score = (2*1+ 1*3 + 0*5) = 5

Do attack again = **True**

So phase 2:

Step 2:

Attacker battle score = (3*1*0.2 + 3*3*0.5 + 1*5*0.8) = 0.6 + 4.5+4 = 9.1

Defender battle score = (2*1*0.9+ 1*3*0.7 + 0*5*0.2) = 1.8 + 2.1 + 0 = 3.9

Step 3:

Attacker:

Artillery_killed = 3,9 % 5 = 0;

artillery.Count = 1 – 0 **=** 1;

Defender battle score = 3.9 - (0 * 5) = 3.9;

cavalry_killed = 3.9 % 4 = 0;

cavalry.Count = 3 – 0 = 3;

Defender battle score = 3.9 - (0 * 3) = 3.9;

infantry.Count = 3 – (3.9 % 1) = 0;

Defender:

Artillery_killed = 9.1 % 5 = 1;

Artillery_killed = 0; // less then *artillery.Count*

artillery.Count = 0 – 0 **= 0**;

Attacker battle score = 9.1 - (0 * 5) = 9.1;

cavalry_killed = 9.1 % 3 = 3;

cavalry_killed = 1; // less then cavalry.Count

cavalry.Count = 1 – 1 = 1;

Attacker battle score = 9.1 - (1 * 3) = 6.1;

infantry.Count = 2 – (6.1 % 2) = -1;

VICTORY!

Attacker has 1 artillery, 3 cavalry and 0 infantry units.

5.2 As state A is surrounded by 2 friendly and 2 enemy states, so 1 artillery, and 2 cavalry move to the State B.

The battle is done.

NOTE: If at any stage the attacked state receives the message that it is under attack from another state, the current battle runs up to step 4, then the attacked state divides the garrison equally, then in the current thread proceeds the step

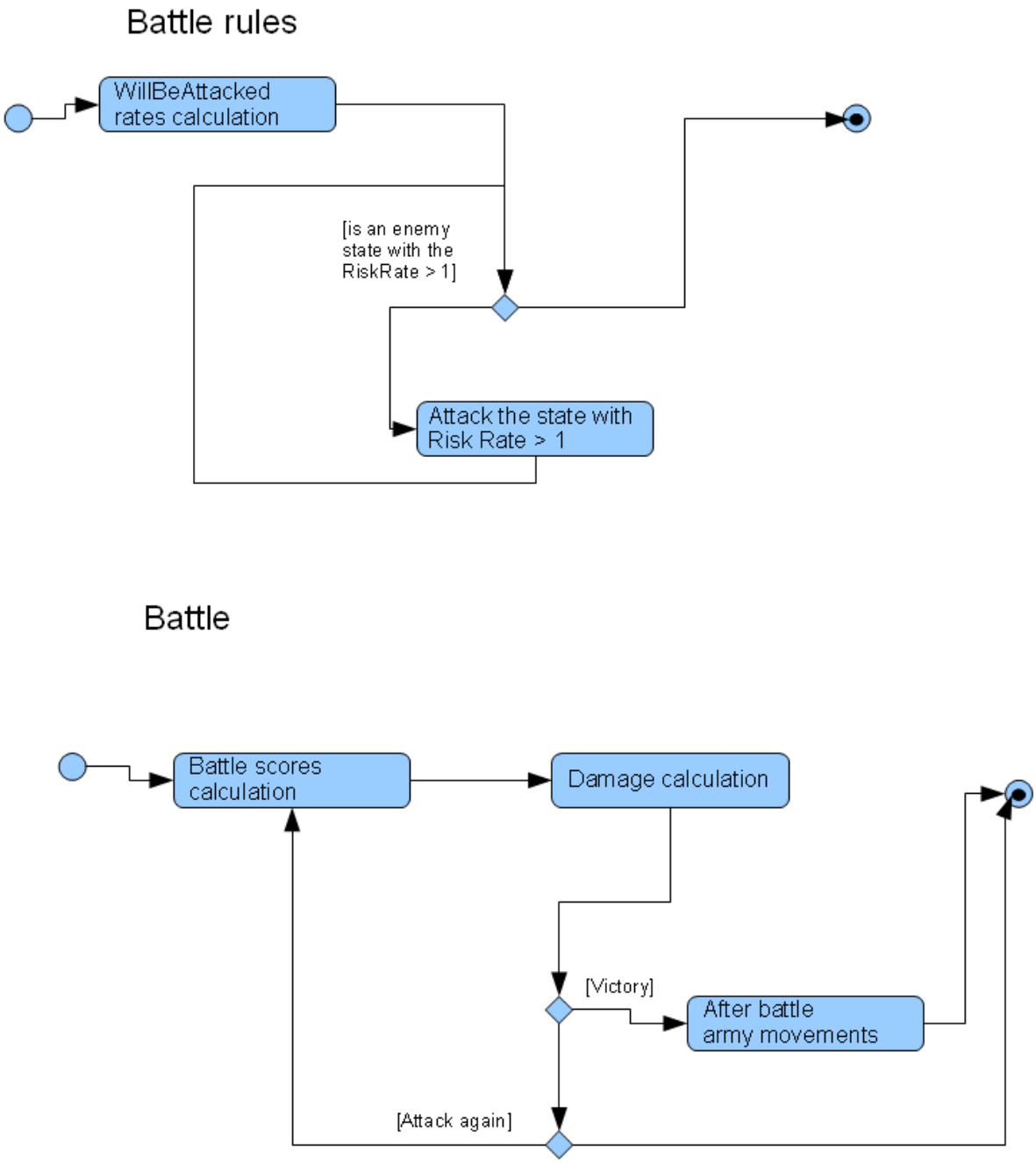

Illustration 2: Battle State Diagrams

Movement Rules

Army movement rules are based on the strength of enemy states, surrounded frontier states.

Example:

State A is surrounded by 3 enemy states (**State E1, State E2,** and **State E3**)

The Basic battle scores are:

Basic defense battle score of **State A**: 41

Basic attack battle score of **State E1**: 20

Basic attack battle score of **State E2**: 5

Basic attack battle score of **State E3**: 30

So, the **State A** risk rate is: (20 + 5 + 30) / 41 = 1.34

And the **State A extra battle score = 41 - 55 = -14**

The movement begins with sorting states ordering by the risk rate.

The movement is one-pass "foreach" process, starting from the state with the minimum risk rate (zero means there are no connected enemy states).

In-circle logic:

```
//sort states ordering by the risk rate
States.SortByRiskRate;

//pass all the states
foreach (State state: States){
        //if there are no extra units do nothing
        if (state.RiskRate < 1){
                // sort connected states ordering by the risk rate
                state.ConnectedStates.SortByRiskRate;
```

```
                    //if the state with the smallest RiskRate has a smaller RiskRate, then
move
                //to that state ALL extra units
                    if (state.ConnectedStates[0].RiskRate > state.RiskRate){
                            state.MoveExtraUnitsTo(state.ConnectedStates[0]);
                    }
            }
}
```

MoveExtraUnitsTo method determinates how many units can be moved (the formula resolves **extra battle scores** back to the units, starting with the most powerful units). The formulas are:

Artillery (if any is available) = **extra battle scores % 6**

Cavalry (if any is available) = the rest of **extra battle scores % 2**

Artillery (if any is available) = the rest of **extra battle scores**

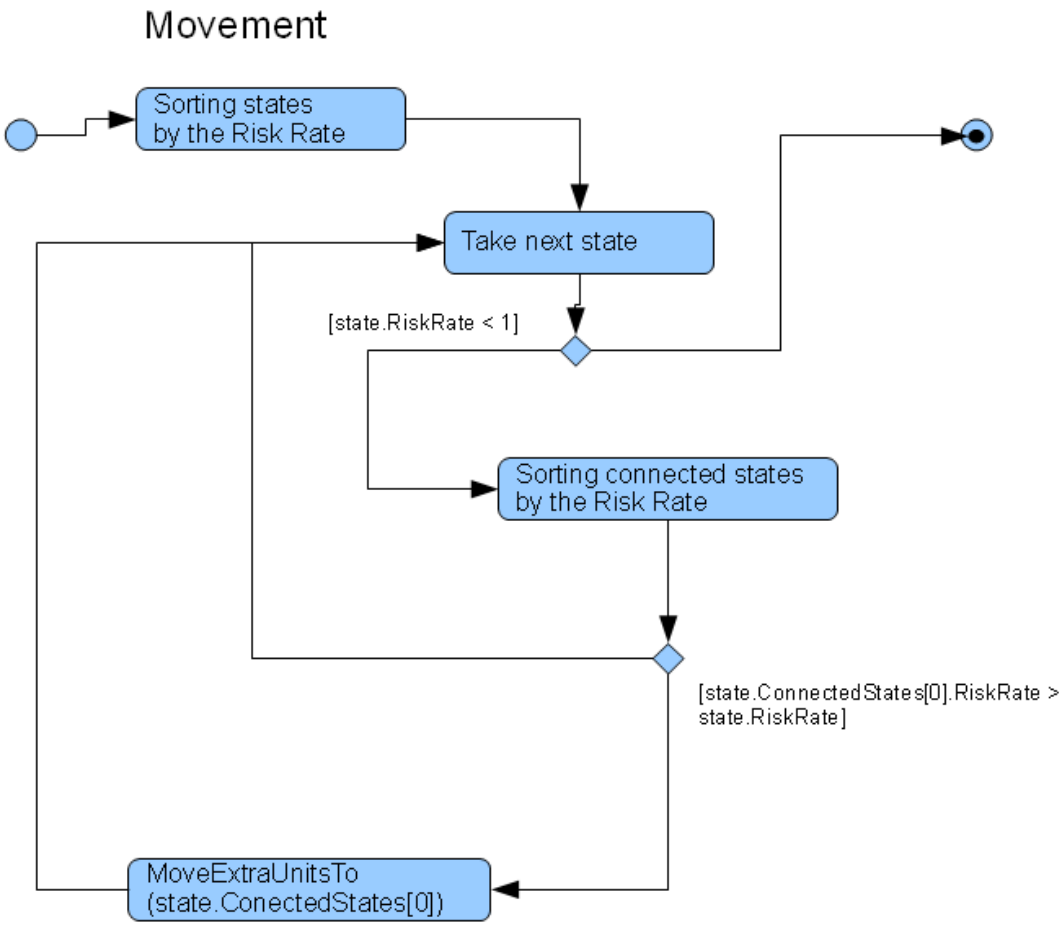

Illustration 3: Movement State Diagrams