



EN SMART MOTSTÅNDARE FÖR BRÅDSPELET RISK

Daniel Westerling

725G51

26/5 2016

Sammanfattning

I detta PM kommer jag presentera applikationen Bert. Jag kommer inleda med att förklara hur jag gått till väga med framtagningen av Bert, för att sedan gå in mer på de tekniska lösningarna som ligger till grund för besluten som Bert tar. Jag kommer även presentera skärmsklipp från gränssnittet både av den färdiga varan, och tidigare versioner av Bert.

Innehåll

| | |
|---|----|
| Sammanfattning..... | 2 |
| Inledning..... | 4 |
| Tillvägagångssätt..... | 5 |
| Gränssnitt..... | 7 |
| Beslutslogik..... | 10 |
| Regioner..... | 10 |
| Antal enheter..... | 11 |
| Nyckelterritorium..... | 11 |
| Ledare..... | 12 |
| Ofullständigheter med nuvarande beslutslogik..... | 12 |
| Reflektion..... | 14 |
| Referenser..... | 15 |

Inledning

Ett flertal gånger har jag reflekterat över hur svårt det är att få tag på en tredje och fjärde person för att till fullo kunna njuta av att spela ett långt parti av strategispelet Risk. Det har lett till flertalet olika spelvarianter för att kunna skapa någon spänning av att endast vara två spelare. Detta har lett till att jag nu valt att ta fram en AI för att vid behov kunna fylla ut de tomma platserna. Resultatet är en fullt fungerande AI som har givits namnet Bert, efter ett troll i J.R.R. Tolkiens bok *The Hobbit, ot There and Back Again*.

Bert är framtagen i första hand för att kunna bemästra en egen framtagen karta för Sagan om Ringen Risk. Denna spelplan skiljer sig inte väsentligt från hur den vanliga planen ser ut, men det finns mindre skillnader i hur var speltur går till. Tanken är dock att koden i sin grund går att använda till liknande varianter av Risk, och de största förändringarna man får göra är att ändra namn och relationerna på de olika territorierna.

Jag har valt att skriva koden helt i java eftersom det är ett språk som jag numer har lärt mig att behärska. Java är dessutom plattformsoberoende. Det innebär att man kan köra samma kod på olika plattformar, förutsatt att det finns en javakompilerare för den valda plattformen.

Mina egna förutsättningar med att skriva kod i Java är de kurser som har tillhanda givits i utbildningen. Därav förekommer det inte referenser till någon typ av litteratur som källa. De gånger då jag har haft utomstående hjälp så har det varit till största delen genom webforumet Stack Overflow.

För att dessutom inte förvirra läsaren med termer såsom; "AI", "Artificiella intelligensen", "datorn", osv. Så har jag valt att i denna skrift använda "Bert" för bestämd form.

Tillvägagångssätt

Jag valde att inte fokusera på gränssnittet i första hand. Jag skrev koden helt och hållet för att kunna köra genom att skriva kommandon i konsolen som ges av Eclipse. Jag såg först till så att denna variant av applikationen fungerade helt problemfritt innan jag började med att skapa gränssnittet. Den största skillnaden mellan dessa varianter, utöver rent grafiskt, är att det går att skriva in betydligt mer felaktiga inputs i den konsolbaserade. På grund av det finns det i den tidiga versionen väldigt mycket felhanteringskod. Något som i stor utsträckning har skalats bort i den senare koden.

```
Console
Launcher (1) [Java Application] C:\Program Files\Java\jdk1.7.0_65\bin\javaw.exe (9 maj 2016 16:08:51)
Is red an AI? Yes [Y] or no [N]?
Y
Is black an AI? Yes [Y] or no [N]?
N
Is yellow an AI? Yes [Y] or no [N]?
N
Is green an AI? Yes [Y] or no [N]?
Y
I am Grima Wormtounge. For die you must.
How many territories do I start with?
3
Write territories below.
THE SHIRE
One unit added on The Shire
ARNOR
One unit added on Arnor
LORIEN
One unit added on Lorien
-----
Grima Wormtounge is ready!
-----
I am Thorin Oakenshield. War is the province of men.
How many territories do I start with?
1
Write territories below.
NURN
One unit added on Nurn
-----
Thorin Oakenshield is ready!
-----
Red [R]. Green [G]. Attack territory [A], Move the Ring [M].
R
-----
Units [U], attack [A], move [M], leader [L], show territories [T], the Ring [R], choose player [C], end turn [E].
U
-----
What will I do? Place units [P], modify units [M], or go back [B]?
P
How many units do I get to place?
6
I'm thinking about putting:
-----
2 ghastly units on The Shire.
4 hideous units to extend my army in Lorien.
-----
Units [U], attack [A], move [M], leader [L], show territories [T], the Ring [R], choose player [C], end turn [E].
A
-----
This glorious army attacks Grey Havens. I go from The Shire with 1 battleborn unit. My men will fight to the end.
Did I win [W] or did I lose [L]?
L
```

Interaktionen med konsolen. Den grönmarkerade texten är input från användare, och den svarta texten är reaktionen ifrån systemet.

Jag började skriva allting i Eclipse. När jag sedan var klar med en fungerande kod för konsolen så gick jag över till Netbeans. Detta då Netbeans har bättre verktyg för att enkelt kunna bygga ett användargränssnitt.

Jag har under programmeringen försökt att till så stor utsträckning som möjligt att använda metoder för återanvändning av kod. Från början var den en väldigt liten del av koden som var återkommande i olika metoder. Detta blev dock betydligt mer i och med mitt beslut att kunna stödja flera datorstyrda AI under samma körning. Det finns ett fåtal metoder som körs av samtliga olika AI. Detta val gjordes för att samtliga metoder redan var fungerande när beslutet togs att flera AI skulle kunna köras samtidigt. Det är därav ingen omöjlighet att en del metoder går att korta ner, eller kombinera mellan olika AI. Detta är definitivt ett jobb om vore aktuellt för en vidareutveckling av applikationen.

För att bygga gränssnittet använde jag mig av swing, som är en verktygslåda för att kunna bygga grafiska gränssnitt med Java. Eclipse har flertalet plugin att välja mellan, men i nuläget finns det ingen på samma nivå som motsvarigheten i Netbeans. Swing kan tendera att ge ett ganska tråkigt och färglöst intryck på användaren. Det ska dock poängteras att mitt huvudsakliga syfte med detta arbete inte vara att lägga mest fokus på gränssnittet, utan snarare att få en fungerande AI.

I swing använde jag mig av något som kallas för en relativ layout. Det innebär att varje element som läggs in på skärmen är relaterad till andra element som redan finns. Exempelvis beror storlek och position på hur de relaterade elementen är placerade. Detta kan leda till att en del knappar och texter inte alltid befinner sig på samma plats.

Gränssnitt

Som tidigare skrivet så använde jag mig av Swing. När man startar programmet bemöts man av en introduktionsfas, där användaren väljer vilka färger som styrs av Bert, samt vilka territorier var färg börjar med. När varje färg har tilldelats territorium hamnar man i huvudmenyn. Här finns alternativ för att påbörja fasen för en AI, attackera ett territorium under Berts kontroll, eller placera ringen på spelbrädet. De två sistnämnda alternativen är de enda två från användaren som påverkar Berts färger, och därför är de med inbyggda i gränssnittet.

Om man väljer en färg så hamnar man i vald färgs huvudmeny, där man väljer vad Bert ska göra med den färgen. Här återfinns samtliga vanliga spelaralternativ som; attackera, förstärk, flytta, mm.

För att alltid hålla uppdaterad om vad som händer, och vad som utförs så förekommer det väldigt mycket dialog i gränssnittet. Det handlar lika mycket om guide från systemet, som dialog från Bert. Jag har särskilt dessa genom att gör Berts dialoger mer unika än systemdialog, som aldrig särskiljer sig. Ett exempel är i attackfasen där Bert genom en kombination av flertalet olika slumpmässiga strängar, för tillfället kan kombinera 713 318 400 olika unika strängar. Detta är tänkt att ge n känsla av att man spelar mot en unik motståndare.

Nedan följer fyra olika på följd genererade strängar för att närmare visa på olikheten i dessa.

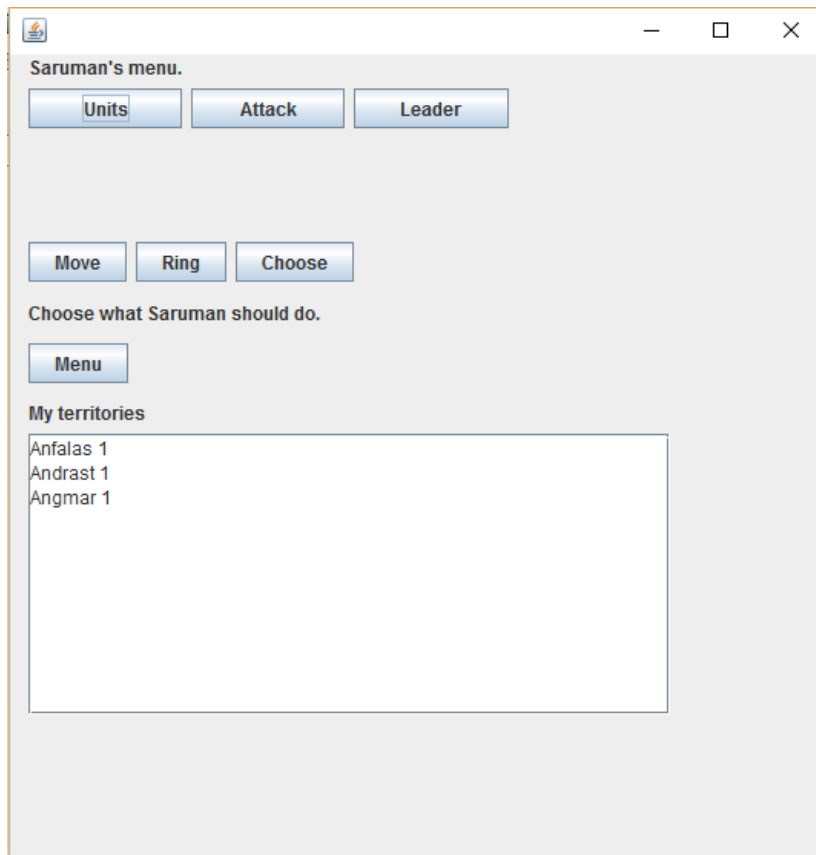
"I march towards Rhudaur from Angmar. I brought 7 ghastly units. My kins will always hunt you".

"I am hungry for flesh. I march towards Forodwaith from Arnor. With me I have 6 battlewor units. No mortal can stop me!"

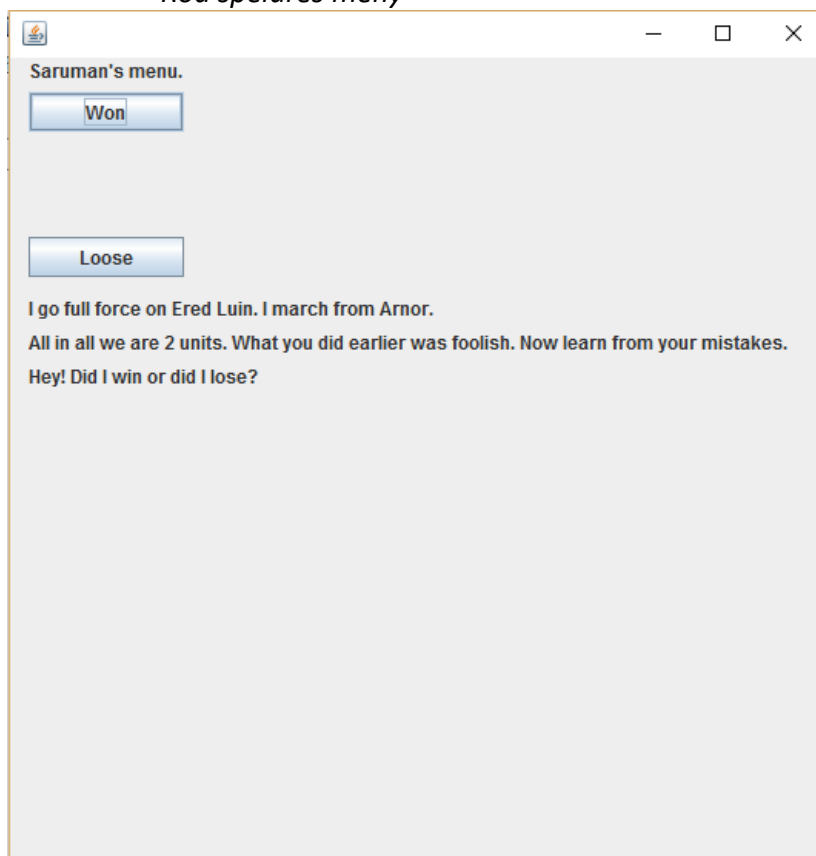
"You are doomed. I attack Druwaith laur. I attack from the neighbouring Anfalas. I have 2 hard units! My name, Smaug. Remember it!"

"Oh my, I march towards Cardolan. Get ready to be attacked by Arnor. Prepare for 6 noble units. Your decisions are, strange!"

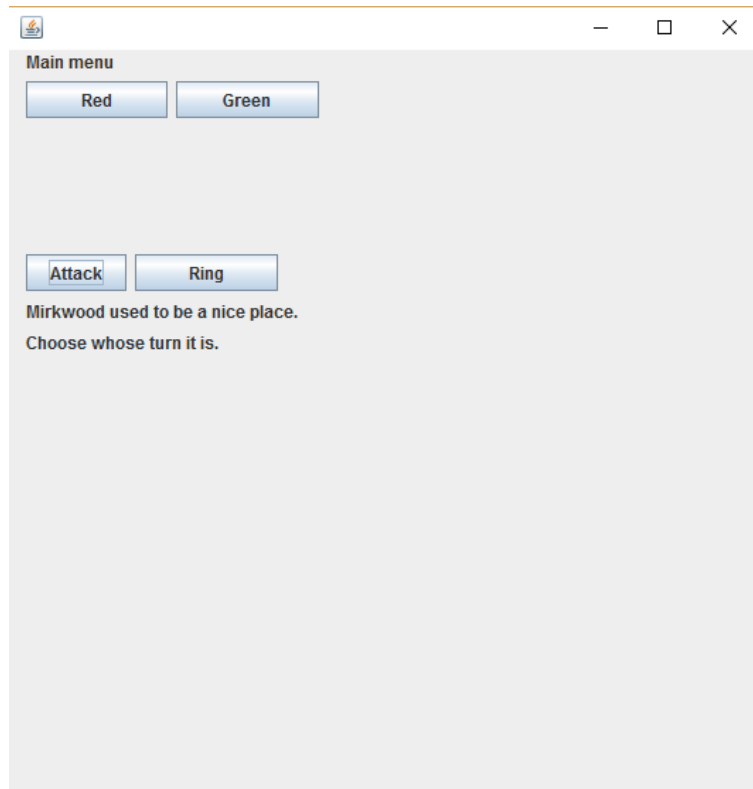
I var tillfälle där det inte påverkar händelser i koden finns det också möjlighet att gå tillbaks till tidigare meny.



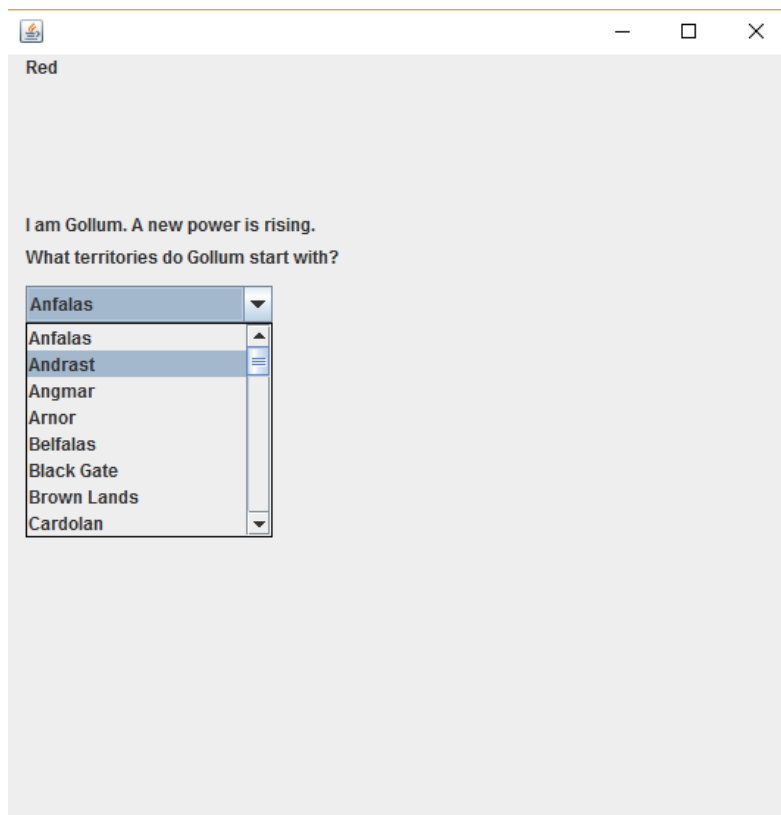
Röd spelares meny



Röd spelares attackfas



Huvudmeny



Val av territorium

Beslutslogik

För att få Bert att bete sig som en verklig motståndare så har jag baserat hans beslut delvis på slumpen, och delvis på ett antal kriterier som gör vissa val mer troliga än andra. Dessa kriterier har jag baserat delvis på till synes självklara logiska beslut, och även på strategi från Garrett Robinsons *The Strategy of Risk* (n.d.). Bert tar alltid egna beslut gällande aktionerna på spelplanen. På grund av händelsekort som förekommer i denna utgåva av Risk, så kan ordningen på spelrundans faser vara annorlunda varje omgång. Därför får man alltid assistera Bert med vad han ska utföra. Hur han sedan utför uppgiften bestämmer han helt själv. De val Bert gör är följande:

- Antal enheter att placera på egna territorium.
- Vilka territorium som ska ta emot enheterna.
- Hur många attacker som ska genomföras.
- Vilket territorium som utför attacken.
- Vilket territorium som ska attackeras.
- Hur många enheter som ska vara med i attacken.
- På vilket territorium ledaren placeras.
- Vilket territorium enheter flyttas från.
- Vilket territorium enheter flyttas till.
- Hur många enheter som ska flyttas.
- Vilket territorium ska Ringen flyttas till.
- Val av motståndare.

Det finns flertalet kriterier som används på olika sätt för att styra Bert till att prioritera de mer självklara valen. Här följer en kortare beskrivning om vilka kriterier som används och vad de innebär.

Regioner

I Risk är spelplanen indelad i ett antal olika regioner, där varje region består av ett antal territorium. Om man innehaver samtliga territorium i en region får man bonusenheter. Med detta i åtanke har jag valt att ett kriterium är att det territorium man ska attackera befinner sig i samma region. Det är alltså större chans att Bert attackerar ett gränsande territorium i samma region, än ett i en annan region.

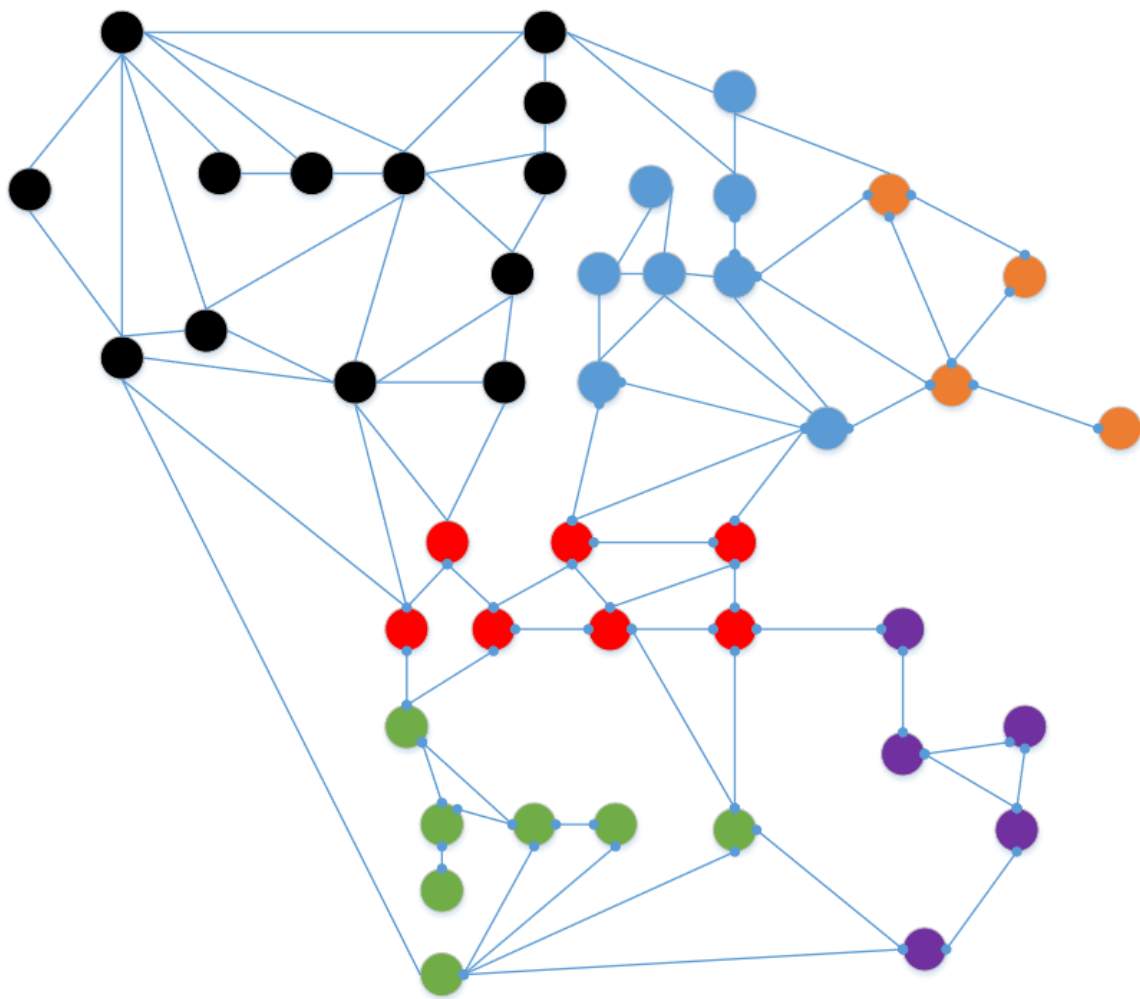
Antal enheter

Detta tycks som en självklarhet att Bert ska fokusera på territorier med mindre enheter än han själv har. Detta kriterium innebär att chansen är dubbelt så stor att Bert attackerar ett territorium med mindre enheter än ett med mer enheter än det som attackeras ifrån. Varför jag fortfarande väljer att ha en någorlunda hög andel attacker från territorium med mindre enheter är för att detta är något man ofta gör i Risk för att exempelvis försvaga motståndarens territorium. Dessutom ska det fortfarande vara möjligt att Bert väljer att attackera med vilket möjligt territorium som helst. Han ser dock på antalet enheter och har med det i beräkningen. Det ska också noteras att det finns regler för att separera vilken typ av övertag det handlar om. Dessa regler innebär att ett överläge med minst tio enheter har tre gånger så stor chans att väljas som ett territorium utan överläge alls.

Bert kontrollerar även själv hur många enheter han anfaller med. Från början var det en väldigt enkel logik där han tar fram ett slumpmässigt tal inom en gräns given av antalet enheter på det attackerande territoriet. Det märktes ganska snart att detta tenderade att innebära flertalet mycket små slag. Därför finns det numer en kontroll för att öka chansen att Bert attackerar med ett större antal enheter. Just nu fungerar den kontrollen följande: Om slumptalet för attackerade enheter är mindre än hälften av antalet möjliga attackerande enheter, så är det 40% chans att antalet attackerande istället regleras till maximalt möjliga. I och med denna ändring har det blivit mer större strider i spelet, och det har även tagit bort en viss känsla av att Bert enbart spelar på slump vid val av antal attackerande.

Nyckelterritorium

Nyckelterritorium är ett begrepp för territorium som har ett vist strategiskt övertag. Detta benämns i Robinsons text i en mer "inbakad" kontext. På ett bräde generaliseras dessa av att de är mindre passager utan alternativa vägar som spelare måste kontrollera för att kunna skärma av, eller skapa kedjor för att kunna transportera enheter. Man vill ofta innehava dessa territorier, och även tenderar dessa i matcher att ha en större mängd enheter på sig än andra. Nyckelterritorier tas i beräkningen för vart enheter kommer placeras ut, vart enheter bör flyttas, och vilka territorier som bör attackeras. Vad som hypotetiskt riskerade att hända med nyckelterritorier var att dessa skulle bli överrepresenterade från Berts sida, och därför var det länge oklart huruvida jag skulle ta med dem eller inte. Jag valde dock att ta med efter att ha testat med och utan, och upptäckt att de utvalda nyckelterritorierna inte alls är överrepresenterade bland Berts aktioner.



Bilden ovan visar samtliga territorium med relationer. De olika färgerna symboliserar olika regioner. Nyckelterritorium kan vara antingen ett territorium med stort antal relationer, eller ett territorium med få relationer, där en relation är ett territorium i en annan region. Vilka som i koden beskrivs som nyckelterritorium är hårdkodat och bygger på både teori och ifrån praktisk kunskap efter spelomgångar.

Ledare

En typ av enhet som står ut från de andra är ledaren. En ledare styrker alltid det territorium som den befinner sig på, eller det under attack. Självklart måste Bert även ha koll på vart han har sina egna, och andra ledare. Bert har större chans att undvika att attackera en ledare, men han har också större chans att attackera med en ledare. Denna spelpjäs är unik för *RISK: Lord of the Rings* och bearbetas inte av Robinson.

Ofullständigheter med nuvarande beslutslogik

Vad som förekommer i nuläget är att Bert har full koll på vart andra AI har sina enheter och ledare. Jag gjorde det valet att inte skriva in var varje användare placerar samtliga sina enheter. Detta är för att det skulle innebära att man som spelare skulle få skriva in och uppdatera Bert för varje litet drag man gör. Den enda gången en spelare nu uppdaterar Bert

under ens tur är när man har attackerat, och ändrat antalet enheter på ett territorium som hålls av någon utav Berts färger. Det skulle medföra en pålastning att skriva istället för att tillföra en enklare motståndare. Som det är nu måste man visserligen assistera Bert med en del, men allt man visar honom berör han själv. Detta medför att dragen som utförs blir mer genomtänkta om det spelar minst två färger som styrs av Bert, då han alltid har full koll på dessa.

Dock ska det nämnas att Bert vet hur många enheter det finns på territorier som har erövrats av spelare, i och med att man skriver in den informationen när man övertagit ett territorium från Bert. Därav kan Bert ibland genomföra attacker baserat på hur många enheter som senast stod på ett sådant nämnt territorium.

Reflektion

Arbetet i sig har varit otroligt roligt att arbeta med. Något som styrker detta är att själv kommer att ha stor nytta av den slutgiltiga produkten. Det ska dock sägas att applikationen inte är riktigt färdig ännu. Det kommer behövas använda flertalet gånger med mindre korrigeringar för att Bert ska kännas som en smart motståndare.

Den vidareutveckling som vore mest intressant är hur man skulle kunna se till att AI alltid har full koll på hur många enheter som befinner sig på territorium som en mänsklig spelare kontrollerar. Som tidigare nämnt så har jag undvikt att låta användaren skriva in denna information, men undantag för när ett AI kontrollerat territorium förloras till en mänsklig spelare. Detta för att undvika att den största speltiden spenderas framför datorn och inte på spelbrädet. Att dock testa vidare på hur man skulle kunna gå till väga för att minska antalet interaktioner med gränssnittet vore en naturlig fortsättning på projektet.

Andra delar som alltid kan utvecklas vore att låta Bert räkna mer på sannolikheter än vad han gör nu. I dagsläget fungerar algoritmen så att han lägger till varje territorium som uppfyller ett kriterium i en lista. Därför kommer territorium som uppfyller fler territorium att läggas till fler gånger. Som koden ser ut i dagsläget så spelar inte enheter så stor roll för Bert som det gör för en mänsklig spelare, och därför vore det intressant att utveckla algoritmerna för just val av territorium. Detta är egentligen ingen stor ändring och kan komma att göras inför presentationen av applikationen.

Jag har inte stöt på några större problem med utvecklandet av Bert. Det ska dock poängteras att kodens estetiska sida mycket väl kan förbättras för att få en bättre kodstandard. Det som kommer att genomföras snarast är att rensa på metoder som inte används, eller som helt enkelt inte behövs. Även så kommer var dialog att läggas i separata textdokument för att spara plats i koden. Det kommer även att behövas läggas till viss felhantering. I dagsläget finns det ingen felhantering för felaktig input av data. Detta hanterades fullständigt i den tidiga konsolversionen, men jag har inte implementerat någon felhantering för felaktig dialoginput. Dock så är det betydligt mindre input i det grafiska gränssnittet än vad det tidigare var.

Det ska dock nämnas att tanken inledningsvis var att skapa en applikation för Android. Detta blev inte av då jag inte hade tid för att skapa ett fungerande gränssnitt för Android. Istället skrev jag koden färdig för att köras på PC i första hand. Dock så är Java plattformsoberoende, vilket innebär att kärnan i koden går att köra på flera platser än PC. Dock så kommer det behövas ett annorlunda gränssnitt för att kunna köra på Android.

Referenser

Strategi

G. Robinson, (n.d.) *The Strategy of Risk*,

Hämtat: 22/4 2016

<http://web.mit.edu/sp.268/www/risk.pdf>

Om RISK

[http://www.hasbro.com/common/instruct/Risk The Lord of the Rings.pdf](http://www.hasbro.com/common/instruct/Risk%20The%20Lord%20of%20the%20Rings.pdf)

Java

<http://stackoverflow.com/>

<http://www.oracle.com/index.html>