

Projektuppgift

DT071G – Programmering i C#.NET

GeoGamr

Konsollbaserat geografispel

Tommi Ojaskivi



Mittuniversitetet

MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. Campus Sundsvall Holmgatan 10, SE-851 70 Sundsvall.

Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.

Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

MITTUNIVERSITETET
Avdelningen för informationssystem och -teknologi

Författare: Tommi Ojaskivi, toj1800@student.miun.se
Utbildningsprogram: Webbutveckling, 120 hp
Huvudområde: Datateknik
Termin, år: HT, 2021

Sammanfattning

Ett geografispel ska skapas med hjälp av C# .NET5. Spelet ska vara en konsollapplikation där användaren ska kunna välja mellan olika spellägen. Ett där användaren ska skriva ut huvustaden för ett specifikt land, ett där användaren ska skriva ut landet för en specifik huvudstad, och ett där användaren ska ange om ett land är större eller mindre än ett annat land. Spelaren kommer kunna välja bland olika svårighetsgrader samt vilka regioner som kan förekomma (Europa, Amerika, Asien, Afrika). Data till spelet läses in från en JSON-fil. Det finns en poänglista med de 10 bästa poängen för vardera spelläge, även dessa sparas i separata JSON-filer. Programmet kommer att skapas i C# och kommer att vara objektorienterat.

Innehållsförteckning

Sammanfattning.....	iii
1 Introduktion.....	1
1.1 Bakgrund och problemmotivering.....	1
1.2 Detaljerad problemformulering.....	1
2 Teori.....	2
3 Metod.....	3
4 Konstruktion.....	4
4.1 Förberedelser.....	4
4.2 Start av spel.....	5
4.3 Spellägen.....	6
4.3.1 Svårighetsgrad.....	6
4.3.2 Matcha huvudstad med land.....	7
4.3.3 Topplistan.....	7
4.4 Sökning av ett land.....	8
4.5 Clear-funktionen.....	9
5 Resultat.....	10
6 Slutsatser.....	11
Källförteckning.....	12
Bilaga A – Wireframe huvudmeny.....	13
Bilaga B – Spelomgång, lätt svårighetsgrad.....	14

1 Introduktion

1.1 Bakgrund och problemmotivering

Ett geografispel ska skapas med hjälp av C# .NET5. Spelet ska vara en konsollapplikation där användaren ska kunna välja mellan olika spellägen. Ett där användaren ska skriva ut huvustaden för ett specifikt land, ett där användaren ska skriva ut landet för en specifik huvudstad, och ett där användaren ska ange om ett land är större eller mindre än ett annat land. Spelaren kommer kunna välja bland olika svårighetsgrader samt vilka regioner som kan förekomma (Europa, Amerika, Asien, Afrika). Data till spelet läses in från en JSON-fil. Det finns en poänglista med de 10 bästa poängen för vardera spelläge, även dessa sparas i separata JSON-filer. Programmet kommer att skapas i C# och kommer att vara objektorienterat.

Målgruppen för spelet är väldigt bred, eftersom det går att justera svårighetsgrad, så personer som är sämre på geografi kan välja en lättare svårighetsgrad och därmed få mer hjälp.

1.2 Detaljerad problemformulering

- Ett geografispel ska skapas i C#
- Koden ska vara objektorienterad
- Data ska läsas in från en JSON-fil
- En poänglista ska sparas till en JSON-fil
- Det ska finnas olika svårighetsgrader
- Det ska vara möjligt att spela endast en specifik region
- Det ska finnas flera olika spellägen
 - Matcha huvudstad med land
 - Matcha land med huvustad
 - Landyta - större eller mindre
- Det ska finnas möjlighet att söka upp information om ett land
- Wireframes och flödesscheman ska skapas
- Korrekt datatyp ska användas i största möjliga mån (int, string, Country etc istället för var)

2 Teori

C# – C# är ett objektorienterat programmeringsspråk som gör det möjligt att skapa många olika typer av applikationer som körs med .NET-runtime. C# härstammar från programmeringsspråket C och är därmed väldigt likt C, C++ och Java. Dessutom är C# typehintat, vilket betyder att det definieras vilken typ varje variabel är. Så om en variabel är en sträng, heltal, eller boolean.[1]

.NET – För att C# ska kunna köras på i princip vilken plattform som helst, används något som kallas för .NET. Det är ett virtuellt exekverings system (Virtual Execution System) och består av en mängd klassbibliotek. När programmet sedan körs så konverterar .NET om programmet från C# kod till enhetens egna språk.
[1]

3 Metod

Innan programmerandet sätter igång så ska wireframes skapas för att få en tydligare bild över hur programmet ska komma att se ut. Dessa wireframes kommer att skapas med webbapplikationen Draw.io. Utöver wireframes skapades även flödes-scheman. Även dessa skapades i Draw.io.

En färdig JSON-fil¹ hämtas från internet med information om alla världens länder. Denna fil görs sedan om med hjälp av ett egenskrivet JavaScript-program. Endast medlemstater i FN (Förenta Nationerna) används i den slutgiltiga filen.

All C# kod kommer att skrivas i .NET 5. JetBrains C# IDE Rider kommer att användas för att skriva all kod samt testköra programmet.

GitHub kommer att användas som versionshanteringssystem.

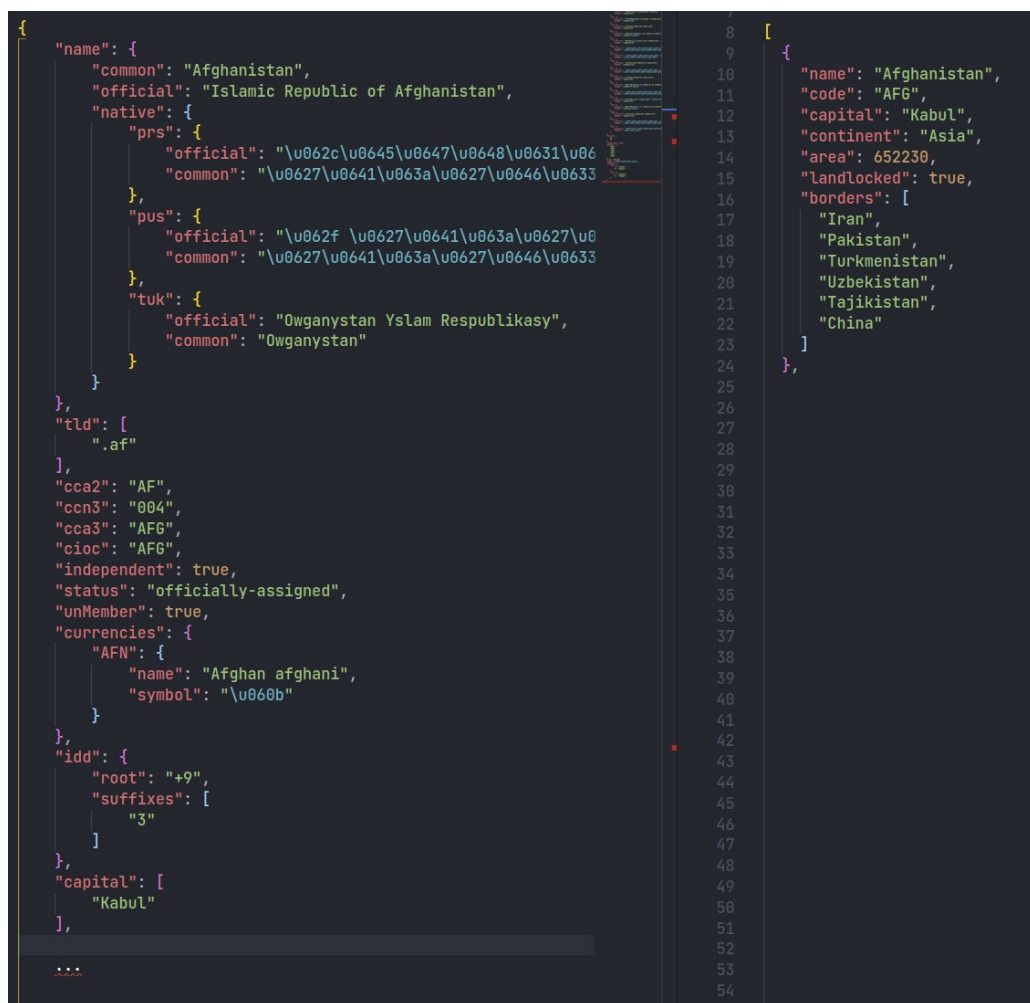
¹ Färdig JSON-fil med alla världens länder: <https://github.com/mledoze/countries>

4 Konstruktion

4.1 Förberedelser

Först hittas det en JSON-fil med all nödvändig data. Informationen som måste finnas tillgänglig är landets namn, huvudstad samt yta. Om det finns mer data så är det bara en bonus. Efter att en passande JSON-fil har hittats skrivs ett simplare JavaScript-program som filtrerar ut vissa länder från listan. Till exempel ska endast FN-länder vara med i spelet, så alla som inte är med filtreras bort. Dessutom rensas mycket av data bort för att göra filen så liten och behändig som möjligt. Huvudstaden är från början i en array, men detta görs om så det endast är en sträng. Sydamerika har egentligen tre huvudstäder, men endast en, Kapstaden, behålls för detta spel.

Den totala filen förkortas ned från 39207 rader till 2507 rader. En minskning med över 90%. Afghanistan förkortas till exempel ned från 167 rader till endast 17 rader (Figur 1).



```
{
  "name": {
    "common": "Afghanistan",
    "official": "Islamic Republic of Afghanistan",
    "native": {
      "prs": {
        "official": "\u062c\u0645\u0647\u0648\u0631\u0627\u062a\u06cc",
        "common": "\u0627\u0641\u063a\u0627\u0646\u06cc\u0633\u062a\u0646"
      },
      "pus": {
        "official": "\u0627\u0641\u063a\u0627\u0646\u06cc\u0633\u0627\u062a\u0646",
        "common": "\u0627\u0641\u063a\u0627\u0646\u06cc\u0633\u0627\u062a\u0646"
      },
      "tuk": {
        "official": "Owganystan Yslam Respublikasy",
        "common": "Owganystan"
      }
    }
  },
  "tld": [
    ".af"
  ],
  "cca2": "AF",
  "ccn3": "004",
  "cca3": "AFG",
  "cioc": "AFG",
  "independent": true,
  "status": "officially-assigned",
  "unMember": true,
  "currencies": {
    "AFN": {
      "name": "Afghan afghani",
      "symbol": "\u060b"
    }
  },
  "idd": {
    "root": "+9",
    "suffixes": [
      "3"
    ]
  },
  "capital": [
    "Kabul"
  ],
  ...
}
```

```
{
  "name": "Afghanistan",
  "code": "AFG",
  "capital": "Kabul",
  "continent": "Asia",
  "area": 652230,
  "landlocked": true,
  "borders": [
    "Iran",
    "Pakistan",
    "Turkmenistan",
    "Uzbekistan",
    "Tajikistan",
    "China"
  ],
}
```

Figur 1: Information om Afghanistan före och efter rensning av data

Wireframes skapas (bilaga A, bilaga B) med webbapplikationen Draw.io. Dessa wireframes står som grund för hur spelet ska se ut när det väl är klart. Efter att wireframes har skapats så skapas även flödesscheman (bilaga C).

Det spånas även över idéer om vilka spellägen som ska implementeras. Till slut fastställs tre olika spellägen. Matcha huvudstad med land, matcha land till huvudstad, och större eller mindre landsyta. Även de olika svårighetsnivåerna bestäms.

På den lättast nivån får spelaren fyra olika alternativ där ett är rätt. På mellannivån får spelaren svarets längd samt varannan bokstav. På den svåraste nivån fås endast längden på huvudstaden. Oavsett svårighetsgrad visas samtliga tecken som inte är en bokstav. För att jämna ut det hela ges mer poäng för rätt svar ju svårare svårighetsgraden är.

4.2 Start av spel

Programmeringssmässigt skapas det först funktionalitet för att läsa in listan med alla länder och tillhörande data. En klass, country (land), skapas. Sedan läses alla länder in från filen (Figur 2). Varje land blir ett nytt Country objekt, och läggs till i en lista.

```
// test if file is found
if (!File.Exists(path:"countries.json"))
{
    Console.WriteLine("Game files missing... can't start. Press any key to exit...");
    Console.ReadKey();
    return;
}

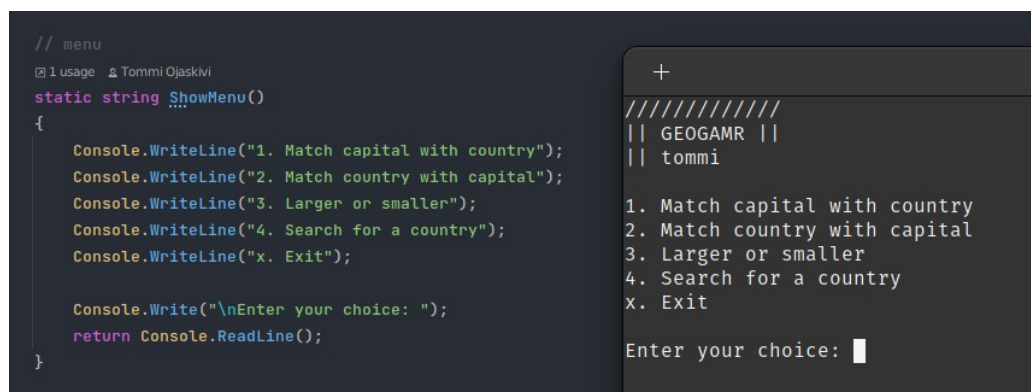
// read in the json file
string json = File.ReadAllText(path:"countries.json");

// deserialize the json into a list of Country objects
List<Country> countries = JsonSerializer.Deserialize<List<Country>>(json);
```

Figur 2: Funktionalitet för att läsa in alla länder till country-objekt

Efter det frågas spelaren om dess namn. Detta namn används sedan eventuellt som namn, om spelaren får tillräckligt höga poäng för att hamna på topplistan, där endast de tio bästa omgångarna sparas.

Spelets huvudmenyn skrivs ut och användaren får mata in ett menyalternativ. I huvudmenyn finns det fem olika alternativ. De tre olika spellägena, möjlighet att visa information om ett specifikt land samt avsluta. I Figur 3 visas koden för menyn samt menyn under programmets körning. Resultatet av användarens inmatning returneras inmatningen till förälderfunktionen. Där används en if-sats och en switch-sats för att kalla på korrekt funktion. Det delas upp i en if-sats och en switch-sats eftersom alla menyalternativ inte kräver samma alternativ. Vill användaren söka efter ett land krävs det varken region eller svårighetsgrad.



```
// menu
1 usage 2 Tommi Ojaskivi
static string ShowMenu()
{
    Console.WriteLine("1. Match capital with country");
    Console.WriteLine("2. Match country with capital");
    Console.WriteLine("3. Larger or smaller");
    Console.WriteLine("4. Search for a country");
    Console.WriteLine("x. Exit");

    Console.Write("\nEnter your choice: ");
    return Console.ReadLine();
}
```

```

+
//////////
|| GEOGAMR ||
|| tommi

1. Match capital with country
2. Match country with capital
3. Larger or smaller
4. Search for a country
x. Exit

Enter your choice: █

```

Figur 3: Kod för meny till vänster och faktiska menyn till höger

Beroende på vilket alternativ användaren väljer så anropas en meny för val av region samt eventuellt en meny för val av svårighetsgrad. Två spellägen (matcha huvudstad med land och matcha land med huvudstad) har region och svårighetsgrad, spelläget där användaren ska avgöra om ett lands yta är större eller mindre än ett annat har endast en svårighetsgrad, men det finns möjlighet att välja region. Genom att välja region kan en användare till exempel välja att endast få frågor om Europa.

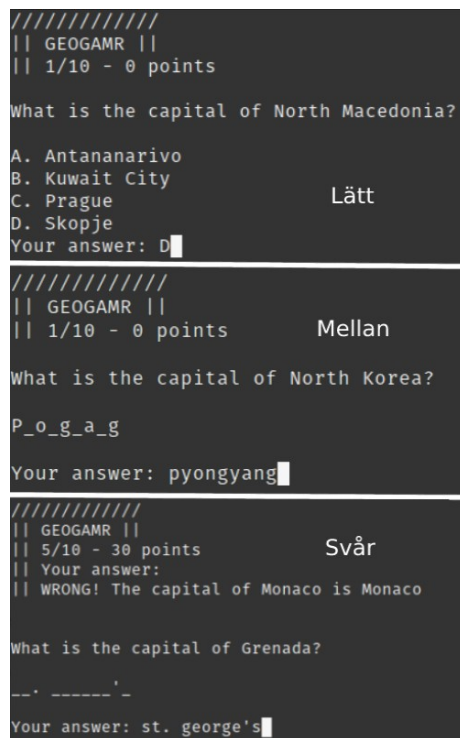
4.3 Spellägen

Varje spelläge har en liknande uppbyggnad. Tio slumpade länder från vald region hämtas följt av att det ställs tio frågor. Frågorna är annorlunda beroende på vilket spelläge som är valt. Vid korrekt svar delas det ut poäng beroende på svårighetsgrad. Om poängen räcker för topp 10 i poänglistan sparas poängen till poänglistan.

4.3.1 Svårighetsgrad

Det finns tre stycken svårighetsgrader att välja mellan. Lätt, medium, svår. På den lättaste nivån ges 10 poäng per korrekt svar, mellan 15 poäng, svår 30 poäng. Vid den lättaste svårighetsgraden får spelaren fyra alternativ att välja mellan. Spelaren kan då antingen skriva in hela svaret eller endast skriva ett av alternativen, A, B, C eller D. Vid mellan-nivån ges första bokstaven samt varannan bokstav. Vid svårt ges endast längden på svaret.

I Figur 4 visas det ett exempel på respektive svårighetsgrad. På mellan och svår skrivs samtliga icke bokstäver ut, så som bindestreck och mellanslag.



```

//////////
|| GEOGAMR ||
|| 1/10 - 0 points

What is the capital of North Macedonia?

A. Antananarivo
B. Kuwait City
C. Prague
D. Skopje
Your answer: D

```

```

//////////
|| GEOGAMR ||
|| 1/10 - 0 points

What is the capital of North Korea?

P_o_g_a_g
Your answer: pyongyang

```

```

//////////
|| GEOGAMR ||
|| 5/10 - 30 points
|| Your answer:
|| WRONG! The capital of Monaco is Monaco

What is the capital of Grenada?

_ _ _ _ _
Your answer: st. george's

```

Figur 4: Exempel på samtliga svårighetsgrader

4.3.2 Matcha huvudstad med land

Varje spelläge har en liknande uppbyggnad. Tio slumpade regioner från vald region hämtas följt av att det ställs tio frågor. Vid korrekt svar delas det ut poäng beroende på svårighetsgrad. Om poängen räcker för topp 10 i poänglistan sparas poängen till poänglistan.

När en spelare väljer spelläget som går ut på att matcha en huvudstad med ett land får användaren först välja region. De olika regionerna som finns är Europa, Asien & Oceanien, Afrika samt Amerika (Nord och syd). Dessutom finns det möjlighet att välja samtliga regioner. I Oceanien, Nordamerika, och Sydamerika finns det relativt få länder, så beslutet togs att slå ihop de med Asian och varandra respektive. Alla kommande frågor kommer att handla om länder som återfinns i vald region. Efter regionvalet får användaren välja svårighetsgrad.

Efter alla alternativ är valda kallas själva funktionen för det valda spelläget.

Först hämtas det hur många poäng användaren ska få per rätt svar. Därefter nollställs poängen, antal korrekta gissningar samt det hämtas alla länder från den valda regionen, följt av att tio slumpade länder från de hämtade länderna hämtas.

Information om hur mycket poäng spelaren har samt vilken fråga den är på skrivs ut högst upp på skärmen med hjälp av funktionen Clear. Därefter itereras det genom de hämtade länderna och totalt 10 frågor ställs, där användaren får svara på en fråga i taget.

Med en switch-sats testas det vilken svårighet spelaren har valt. Om spelaren har valt lätt hämtas det alternativ som denna kan se. Det korrekta svaret samt tre andra alternativ. Mer om svårighetsgrad finns att läsa om i kapitel 4.3.1.

Om ett svar innehåller diakritiska tecken², så godkänns det om en användare gissar på grundbokstaven. Så om svaret är Colombias huvudstad, Bogotá, så godkänns Bogota som svar.

Efter varje svar får användaren se om den svarat rätt eller fel, samt vilket korrekt svar är.

När samtliga tio frågor är besvarade redovisas användarens resultat samt spellägets poänglista skrivs ut. Om en användare har spelat på alla regioner sparas det till topplistan. Sedan rensas topplistan till endast de tio bästa poängen. Om en spelare väljer att spela på endast en specifik region, så visas endast ett meddelande som meddelar att en användare måste spela alla regioner för att ha en chans att hamna på topplistan.

4.3.3 Topplistan

Efter en runda så visas spellägets topplista. Funktionen för att spara och visa topplistan tar tre parametrar. Spelets namn, ett poäng-objekt, samt vald region.

² Diakritiska tecken är bokstäver med accenter, som till exempel á, ñ, â, å, í etc.

I metoden kollas först vilken topplista som ska visas. En lista skapas som ska innehålla de olika poäng-objekten. Sedan testas det om topplistan redan existerar eller inte. Om den inte existerar så skapas den och poängen läggs till. Om poänglistan redan existerar så hämtas samtliga poäng från listan, Därefter skrivs hela poänglistan ut. Totalt finns det upp till tre stycken topplistor. En för vardera spelläge.

Sedan sker en kontroll för att se om spelomgången har skett med alla regioner, om så är fallet så sparas topplistan till en JSON-fil. Annars visas endast ett meddelande som säger att spelaren måste spela på med alla regioner för att spara sina poäng till topplistan.

4.4 Sökning av ett land

En användare har även möjlighet att söka upp information om ett land. Om användaren väljer val 4 i huvudmenyn, uppmanas den att skriva in ett landsnamn som ska sökas efter.

Efter att användaren har matat in ett namn kallas funktionen ShowCountry. Om användaren matar in "random" (slumpmässigt på engelska) visas ett slumpmässigt utvalt land. Om ett land inte hittas visas ett felmeddelande där det står att inget land hittades. Skulle landet hittas i listan skrivs all information om landet ut på skärmen (Figur 5).

```
//////////  
|| GEOGAMR ||  
|| SWITZERLAND  
  
Switzerland - CHE  
The capital is Bern  
The surface area is 41284 km2  
It's located in Europe  
It's a landlocked country  
  
Switzerland borders with 5 countries  
- Austria  
- France  
- Italy  
- Liechtenstein  
- Germany  
  
Press any key to continue...  
█
```

Figur 5: Resultat efter sökning på Schweiz

4.5 Clear-funktionen

En egen version av C# inbyggda `Console.Clear()`-funktion implementeras. Den rensar skärmen från alla tecken så den blir ren. När den egenskapade Clear-funktionen kallas på så körs först den vanliga `Console.Clear()`. Därefter skrivs det ut en slags header för spelet, med lite snedstreck samt spelets namn.

Funktionen har en valfri parameter som är en sträng. Med denna parameter är det möjligt att skriva ut information till skärmen. Här skrivs det bland annat ut information om den pågående rondens så spelaren kan se hur många frågor det har svarats på samt hur mycket poäng som är insamlat just nu. Det skrivs även ut om förra frågan besvarades korrekt eller inkorrekt, samt vilket det korrekta svaret var.

5 Resultat

Ett geografispel (Figur 6) har skapats i C#. Koden bakom programmet är skrivet med objektorienterad programmering. Var har undvikits i största möjliga mån, och istället har korrekt datatyp angetts direkt.

Spelet hämtar all data från en JSON-fil och läser in det till en lista. När en spelare spelar en tillräckligt bra runda för att hamna på topplistan, sparas poängen till en JSON-fil innehållandes alla poäng. Efter avslutat spel skrivs även poänglistans samtliga poäng ut till skärmen.

Det finns tre olika svårighetsgrader att välja mellan. Lätt, mellan och svår. Lättare svårighetsgrad har mest hjälpmedel, men ger även färre poäng vid korrekt svar.

Innan spelets början har användaren möjlighet att välja vilken region den vill spela med. Europa, Afrika, Amerika samt Asien och Oceanien finns att välja mellan.

Totalt finns det tre olika spellägen att spela. Matcha huvudstad med land, land med huvudstad samt större eller mindre yta.

Om en användare vill, så har den möjlighet att söka efter ett specifikt land för att få ut mer information om detta. Detta går inte att göra under spelets gång, utan måste väljas innan spelet börjar.

Innan programmerandet började skapades det wireframes och flödesscheman för att få en tydlig grund som programmeringen ska utgå ifrån.

```
//////////  
|| GEOGAMR ||  
|| 6/10 - 50 points  
|| Your answer: london  
|| CORRECT! The capital of United Kingdom is London  
  
What is the capital of Burundi?  
  
A. Gitega  
B. Kinshasa  
C. Yerevan  
D. Budapest  
Your answer: a
```

Figur 6: Skärmdump på det färdiga spelet

6 Slutsatser

Till slut vart det klart! Det tog ett tag för mig att komma på vad jag skulle komma på för projekt men till slut kom jag på att jag kunde skapa ett geografispel som hämtar data från en JSON-fil och sen skriver till JSON-fil om jag implementerar en topplista.

Det har varit kul att köra C# nu eftersom att jag körde lite C++ några år sedan, och de är "riktiga" programmeringsspråk, till skillnad från JavaScript och PHP. Mycket att tänka på och det är "svårare" att skriva jämfört med t.ex. JavaScript.

Om jag haft mer tid skulle det ha varit kul att implementera lite fler spel. Oändligheterna är, i princip, oändliga. Med mer data är det möjligt att bara lägga till fler spel. Till exempel spel som handlar om befolkningsmängd etc. Under tiden som jag skrev denna rapport kom jag på ett nytt spelläge, försök navigera från ett land till ett annat genom alla länder. Så om man t.ex. börjar i Norge och ska till Tyskland. Vad är den kortaste vägen, genom landsgränser, att hitta till Tyskland? Rätt svar borde vara Norge → Ryssland → Belarus → Polen → Tyskland. (Man kommer inte från Sverige till Danmark via landsgränser)

Jag blev extra nöjd med alternativen för den lätta svårighetsgraden samt hur min egna Clear()-funktion fungerar med möjligheten att skriva ut ett meddelande på skärmen.

Annars tycker att det har gått bra och jag blev nöjd med spelet. Jag tycker att det är väldigt kul med geografi och huvudstäder allmänt men är relativt dålig på huvudstäder utanför Europa, så nu är det bara att nöta det flera timmar om dagen så att det sitter perfekt!

Ha de gött!

Källförteckning

- [1] Microsoft, "A tour of the C# language", <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> Publicerad 2021-11-30 Hämtad 2022-01-23

Bilaga A – Wireframe huvudmeny

```
~~ GeoGamr ~~  
  
1. Capital to country  
2. Country to capital  
3. Larger or smaller  
4. Search country  
x. Exit  
  
Your choice: 3
```

Bilaga B – Spelomgång, lätt svårighetsgrad

-- GeoGamr --

4/10 - 20 points

What is the capital of Sweden?

- A. Stockholm
- B. Oslo
- C. Copenhagen
- D. Helsinki

Your answer:

Bilaga C – Flödesschema

Flödesschema - GeoGamr

