**Applikationsprogrammering i Python, 5DA000**

**Memory Spel, HT18**

**Uppgifts nummer: 1**

**Namn: Andreé Johnsson, Hampus Silverlind**

**CSid: XXXXXXXXX, bio15hsd**

**Datum: 2018-11-30**

**Lärare: Thomas Johansson**

**Problemspecifikation**

Denna uppgift gick ut på att genomföra ett projekt inom en grupp med mål att skapa ett valfritt program. De enda kraven som ställdes i projektspecifikationen var följande:

- Programmet ska ha ett grafiskt gränssnitt som inkluderar filer på något sätt.

- Lösning till programmet ska någon sorts objektorientering.

Utöver de ovanstående kraven så fick alla grupper fritt välja vad för sorts typ av program de ville skapa. I projektspecifikationen så gavs förslag baserat på programinriktning hos dem som läser kursen men även generella förslag som t.ex. spel, som just denna grupp valde. Spelet som vi valde att skapa är baserat på ett äldre fysiskt memory spel var fyra olika färgade knappar blinkar i en viss ordning som användaren sen ska härma. Vi valde att även lägga till en extra funktion till programmet i form av en s.k ”Highscore” där användaren kan lagra sina försök. Eftersom specifikationen för uppgiften inte var specifik just för vårt spel gav det oss fria händer till hur vi skulle utforma det, t.ex. hur svårigheten i spelet skulle behandlas. Kraven vi själva satt på spelet var följande (Utöver de givna):

- Det ska finnas någon sorts informations ruta när programmet startas som hjälper användaren förstå spelet.

- Svårighetsgraden på spelet ska öka linjärt.

- En grafisk nedräkning ska ske mellan varje nivå.

- Det ska vara en bra övergång från när programmet utför saker till när användaren ska göra saker, och vice versa.

- När användaren förlorar i spelet ska hen få möjlighet att lägga in sitt namn och högsta nivå till en highscore fil som även kan visas i programmet.

- Olika funktioner i programmet ska delas upp i sina egna separata klasser som ska kallas från huvudprogrammet.

**Åtkomst och användarhandledning**

Alla filer till programmet ska lagras i en mapp tillsammans, beroende på var mappen lagras måste användaren hänvisa dit i kommandotolken. Själva programmet är uppdelat i flera klasser som ligger i separata filer, beskrivna nedan:

*RGBY-ColorGame\_3.0*: Själva huvudprogrammet som skapar fönstret som innehåller alla funktioner och knappar. Filen innehåller även de flesta popups som dyker upp under programmets gång. **Det är denna fil som användaren ska starta för att spela spelet.**

*1,2,3timer*: Bilder som används för den grafiska nedräkningen.

*Countdown*: En grafisk nedräkning före nivån startar som använder bilderna nämnda ovan.

*ButtonFrame*: Skapar och innehåller knapparna som blinkar och användaren sen klickar på.

*LabelFrame\_S*: Innehåller meny knapparna samt information om nuvarande nivå och liv.

*ScoreFrame*: Ruta som syns när användaren förlorar där hen får fylla i sitt namn och spara sin framgång.

*HighscoreFrame:* Visar användaren den nuvarande highscore listan.

*ScoreSheet*: Innehåller funktioner för att ändra och hantera highscore filen.

*top10scoreSheet*: Själva highscore filen som lagrar användarnas avklarade nivåer.

Som sagt ovan så är det *RGBY-ColorGame\_3.0* som är den enda filen som ska köras av användaren för att spela spelet. De andra filerna är endast där för att dela upp programmet i lättare att förstå delar. När själva spelet körs så förklaras det väldigt noga hur användaren ska manövrera programmet. Nästintill varje steg i spelet startas av användaren genom popups för att se till att användaren är beredd och fokuserad.

**Algoritmbeskrivning**

Programmet har ej några större komplicerade algoritmer utan är mer fylld av grafiska lösningar. De större algoritmerna som behövde lösas var följande:

Blink algoritmen:

* 1. Blinka slumpmässigt baserat på nivån.
  + 1.1 Tar in en förbestämd slumpad sekvens.
  + 1.2 Beroende på elementet i sekvensen blinkar en korresponderande knapp.
  + 1.3 Mellan varje blinkning är en timer.
  + 1.4 Återupprepas tills sekvensen är slut.
  + 1.5 En popup visas.

Bytande av bilder i countdown:

* 1. Upprepa tills nedräkningen når 0.
  + 1.1 Ta nuvarande nummer.
  + 1.2 Baserat på numret byt bild.
  + 1.3 Minska numret med 1.
* 2. Nedräkningen når 0.
  + 2.1 Nedräkning tas bort.
  + 2.2 Blinkningen av knapparna startas.

Starta om nivån:

* 1. Kolla hur många liv användaren har.
* 2. Om användren fortfarande har kvar liv.
  + 2.1 Popup med information.
  + 2.2 Minska antalet liv.
  + 2.3 Ändra antalet liv som syns i fönstret.
  + 2.4 Kalla på nedräkningen för att starta blinkningen.
* 3. Om användaren har slut på liv.
  + 3.1 Popup med information.
  + 3.2 Göm knapparna som blinkar.
  + 3.3 Minska nivån.
  + 3.4 Öppna fönstret för användaren att lägga in sitt namn i highscore.
  + 3.5 Ge valet för användaren att starta om spelet.

Algoritmen för nästa nivå är väldigt lik den för att starta om nivån som är beskriven ovan.

**Lösningens begränsningar**

Den största begränsningen som stöttes på under projektet var kunskapen om alla funktioner som existerar i tkinter. Blinkningen av knapparna löstes enkelt tack vare den inbyggda *flash* funktionen i tkinter. Programmet skulle möjligtvis kunnas göras mer felsäkert och snabbare med mer kunskap runt tkinter.  
En stor begränsning var att få alla saker att passa ihop inom fönstret i en satt struktur. Den grafiska nedräkningen hade önskats att ha som en overlay över knapparna men det gick ej att lösas så en kompromiss valdes.

**Diskussion/reflektion**

Det största problemet med hela projektet var som nämnt i *Lösningens begränsningar* var vår kunskap gällande GUIs. Vi hade svårt att förstå hur GUIn fungerar när det gäller ordningen av funktioner som används. I början när vi försökte lösa blinkningen av knapparna så kördes hela blinkningen igenom utan att knapparna syntes, först när den var klar dök knapparna upp. Att programmera en GUI är annorlunda från tidigare programmering gällande hur koden man skriver läses in av programmet.  
Övergripande var projektet väldigt lärorikt som en introduktion till programmering av GUIs. Det kommer vara mycket man kan ta från detta projekt till de framtida uppgifter på kursen och även utanför till framtida kurser.