2. Arkitektuell översikt

Detta dokument specificerar arkitekturen av spelet Pacman genom följande översikter:

Användarfallsöversikt (Use case View):

Målgrupp: Samtliga delaktiga aktörer i projektet, slutanvändaren inkluderat.

Avsett område: Avsnittet beskriver de användarfall som utgör en betydande del för funktionen av det färdiga spelet. Detta avsnitt beskriver även de förväntningar som slutanvändaren kan tänkas ha på det färdiga spelet (Usecase bild från bouml, utökade use case beskrivningar som vi gjorde till lab 3).

Teknisk lösningsöversikt (Logical view, Data view) :

Målgrupp: Systemdesigners och utvecklare.

Avsett område: Funktionella krav på spelet: Visar hur man har för avsikt att realisera de krav som specificerats på spelet genom användarfallen och av uppdragsgivaren. I detta avsnitt förklaras även hur spelets komponenter interagerar med varandra samt vilka konventioner som gäller vid designen av systemet. (UML, DFD, Sekvensdiagram, Avsnitt om felhantering, paketnamngivning etc.)

Distributionsöversikt (Deploy):

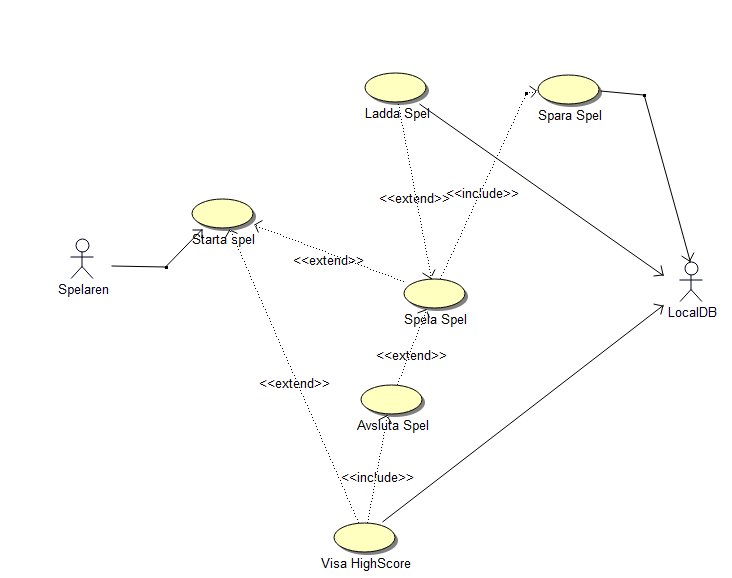
Målgrupp: Distributörer av den digitala produkten

Avsett område: Går igenom hur systemet är tänkt att distribueras och vilka krav som ställs på den aktuella miljön (driftaspekter, Deploy diagram).

2.1 Use-Case View

Bilden visar en överblick på de Use-Cases vi har i vår applikation där Spelaren är vår Actor och LocalDB är de externa aktörerna. Att kommunikationen mellan spelare och databas fungerar felfritt är absolut prio - annars kan inga spel skapas, laddas eller sparas. Om förbindelsen inte är aktiv är menyn det enda som kommer kunna visas och systemet kommer inte kunna utföra något av valen.

Man kan även se var och vilka Use-Cases som inkluderar varandra i de olika scenariers som finns, men även de som exkluderas (inte är beroende av ett tidigare case).



2.2 Use-Case Realization

Här beskrivs några av de Use-Cases som finns tillsammans med tillhörande Story. Ett gemensamt krav som finns är bl.a. att det måste finnas tillräckligt med RAM för att spelet ska fungera korrekt.

|  |  |
| --- | --- |
| User story 1 | Som användare vill jag komma till en huvudmeny där jag kan göra val när spelet startar. |
| Acceptanskriterier | 1. Innan huvudmenyn startar ska företagets logotyp visas. 2. När huvudmenyn körs ska musik spelas i bakgrunden 3. Följande val måste finnas tillgängliga för användaren: Starta spel, Ladda spel, Titta på highscore och Avsluta spel |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Starta spelet | | |
| Created By: | Marcus | Last Updated By: | Kjell |
| Date Created: | 2018-05-11 | Date Last Updated: | 2018-05-12 |

|  |  |
| --- | --- |
| Actor: | Spelaren |
| Description: | Användaren startar applikationen och huvudmenyn visas med alternativ att starta spelet, ladda ett befintligt spel eller titta på highscore listan. |
| Preconditions: | * Tillräckligt med RAM ledigt för att spelet ska kunna fungera. |
| Postconditions: | När spelaren har gjort ett val kommer spelaren att slussas vidare till nästa del av programmet, eller avsluta detsamma. |
| Priority: | Hög |
| Frequency of Use: | Varje gång spelet ska startas samt när spelaren väljer att spara eller avsluta sitt spel från use caset ”spela spel”. |
| Normal Course of Events: | 1. Spelaren startar applikationen. 2. Titelskärmen visas med de alternativ som finns tillgängliga för spelaren. 3. Applikationen inväntar input från spelaren 4. Applikationen exekverar funktion utifrån användarens val. |
| Alternative Courses: | - |
| Exceptions: | 2.EX1. Spelaren försöker ladda ett spel men programmet upptäcker att sparfilen har blivit korrupt. Spelaren ska då få ett felmeddelande och återvända till huvudmenyn. |
| Includes: | Spela spel, Ladda spel, Avsluta spel, Titta på highscore |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |
| --- | --- |
| User story 3 | Som användare vill jag kunna se high score så att jag kan jämföra mig själv med tidigare spelomgångar. |
| Acceptanskriterier | 1. Från huvudmenyn ska valet ”visa highscore” finnas. 2. Så fort ett spel har spelats ska poängen från sessionen sparas i en databas för poäng. 3. När man väljer ”visa high score” så visas poängen från databasen. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | Visa highscore | | |
| Created By: | Mikael | Last Updated By: |  |
| Date Created: | 2018-05-13 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Spelaren |
| Description: | Användaren hämtar hem data från databasen |
| Preconditions: | - Det finns data att hämta.  - Uppkopplingen mellan användaren och databasen lyckades. |
| Postconditions: | Användaren får en visuell översikt av det senaste 10 rekorden. |
| Priority: | Låg |
| Frequency of Use: | - |
| Normal Course of Events: | 1. Spelaren startar programmet.  2. Väljer menyvalet “Show highscore”  3. Applikationen visar en lista med rekord. |
| Alternative Courses: |  |
| Exceptions: | 1. En uppkoppling mellan databasen och användaren kunde inte etableras.  2. Användaren försöker att hämta data som inte finns. |
| Includes: | Titta på highscore |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |
| --- | --- |
| User story 4 | Som spelare vill jag kunna spara spelet för att ändringar/framsteg inte ska gå förlorade. |
| Acceptanskriterier | 1. Spelaren ska kunna komma åt menyn via tilldelad tangent för detta. 2. Där ska val som Spara spel, Pausa och Avsluta spel finnas. 3. Det ska dyka upp en textrad som meddelar när spelet är sparat så spelaren vet när processen är klar. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | Spara spel | | |
| Created By: | Jessica | Last Updated By: |  |
| Date Created: | 2018-05-14 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Spelaren |
| Description: | Användaren sparar alla ändringar som har skett efter den senaste sparade sessionen. |
| Preconditions: | - Det finns ändringar/framsteg att spara.  - Tillräckligt med RAM ledigt för att spelet ska kunna fungera. |
| Postconditions: | När användaren har sparat spelet så kan denna fortsätta spela eller välja att avsluta. |
| Priority: | Medel |
| Frequency of Use: | Varje gång spelaren vill spara sin session eller om Autosave är aktiverat. |
| Normal Course of Events: | 1. Spelaren trycker på den tangent som är avsedd för Meny.  2. En vy visas med tillgängliga val.  3. Spelaren väljer ”Spara spel”  4. Applikationen exekverar funktionen ”Spara spel”. |
| Alternative Courses: |  |
| Exceptions: | 1. Spelet kan inte skriva ändringarna till den befintliga filen pga. att den har kraschat/blivit korrupt – i så fall får spelaren ett felmeddelande. |
| Includes: | Spara spel |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |
| --- | --- |
| User story 7 | Som användare vill jag kunna avsluta spelet ifall jag har behov av det. |
| Acceptanskriterier | 1. Spelaren ska kunna komma åt menyn via tilldelad tangent för detta. 2. Där ska val som Spara spel, Pausa och Avsluta spel finnas. 3. Det ska dyka upp en textrad som meddelar när spelet är avslutat. 4. Spelet avslutas. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 7 | | |
| Use Case Name: | Avsluta spel | | |
| Created By: | Kjell | Last Updated By: |  |
| Date Created: | 2018-05-17 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Spelaren |
| Description: | Användaren avslutar spelet. |
| Preconditions: | - Ett spel är startat och igång.  - Tillräckligt med RAM ledigt för att spelet ska kunna fungera.  - En knapp för att avsluta spelet. |
| Postconditions: | När användaren har pausat spelet kan denna välja att fortsätta spela igen. |
| Priority: | Medel - Hög |
| Frequency of Use: | Varje gång spelaren vill avsluta spelet. |
| Normal Course of Events: | 1. Spelaren trycker på den tangent som är avsedd för Meny.  2. En vy visas med tillgängliga val.  3. Spelaren väljer ”avsluta spel”  4. Applikationen exekverar funktionen ”avsluta spel”. |
| Alternative Courses: |  |
| Exceptions: | 1. Spelet stannar inte den aktuella sessionen utan fortsätter vara aktiv i bakgrunden så spelaren kanske dör eller missar något innehåll/event. |
| Includes: | Avsluta spel |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

2.3 Logical View

Standarder för projektet.

Paketnamn ska inledas med com.ffs.\*

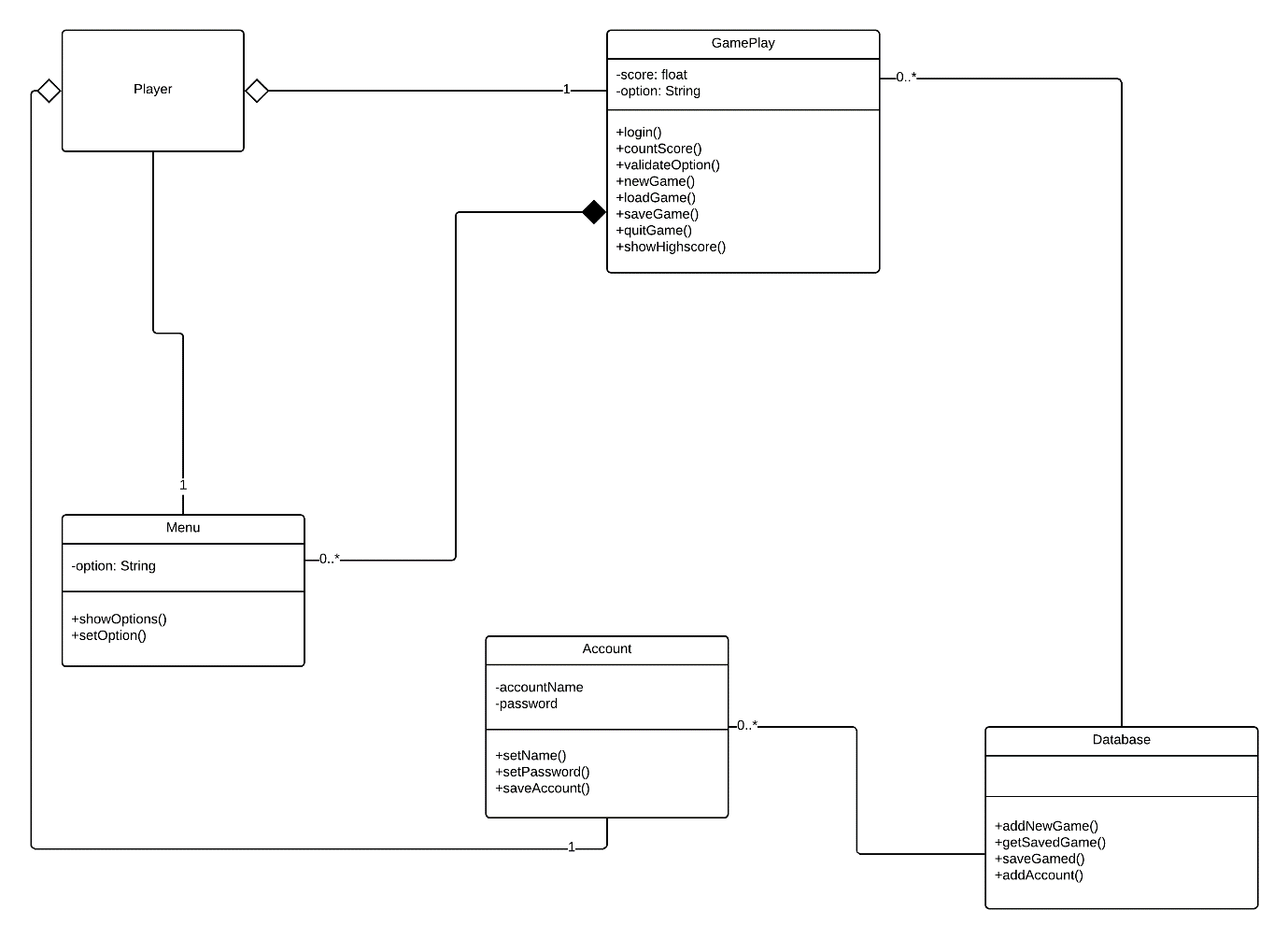
Kodformatering i Eclipse ska ske enligt följande [mall](http://git.eclipse.org/c/jetty/org.eclipse.jetty.admin.git/tree/jetty-eclipse-java-format.xml).

För att genomföra de grundläggande enhetstester som krävs skapas ett separat Java projekt som refererar till JUnit5 och de klasser från spelet som ska testas.   
Testklasserna ska ha samma namn som de klasserna det avser att testa, med tillägget ”Test” på slutet. Testklasserna ska även paketeras på samma vis som de är paketerade i det skarpa projektet, med tillägget unittest i slutet.

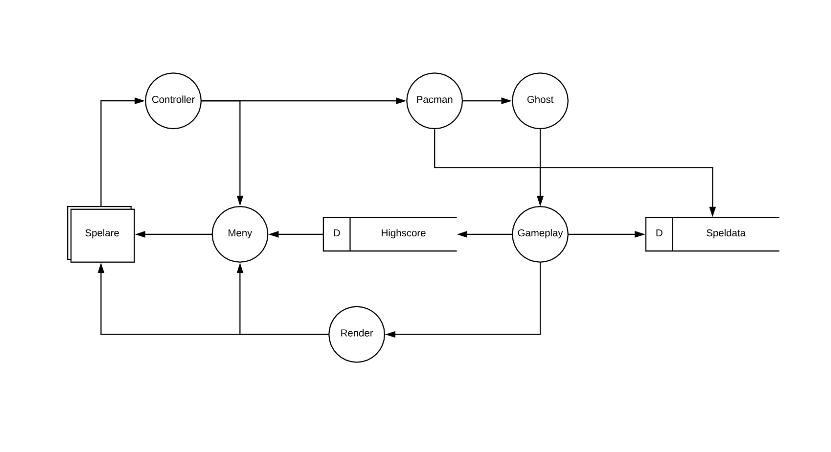
Om egna klasser för felhantering behöver skapas ska dessa läggas i paketet util.exceptions.x.y där ”x” står för den klass och som använder sig av det aktuella falmeddelandet och ”y” vilken metod som avses.

Programöversikt.

I bilden nedanför illustreras några av de funktioner och klasser som bygger systemet, även relationer dessa emellan. Några klasser är beroende av andra för att de ska kunna existera, medans andra instansieras oavsett.



Nästa bild visar istället hur data skickas mellan de olika objekten/enheterna.

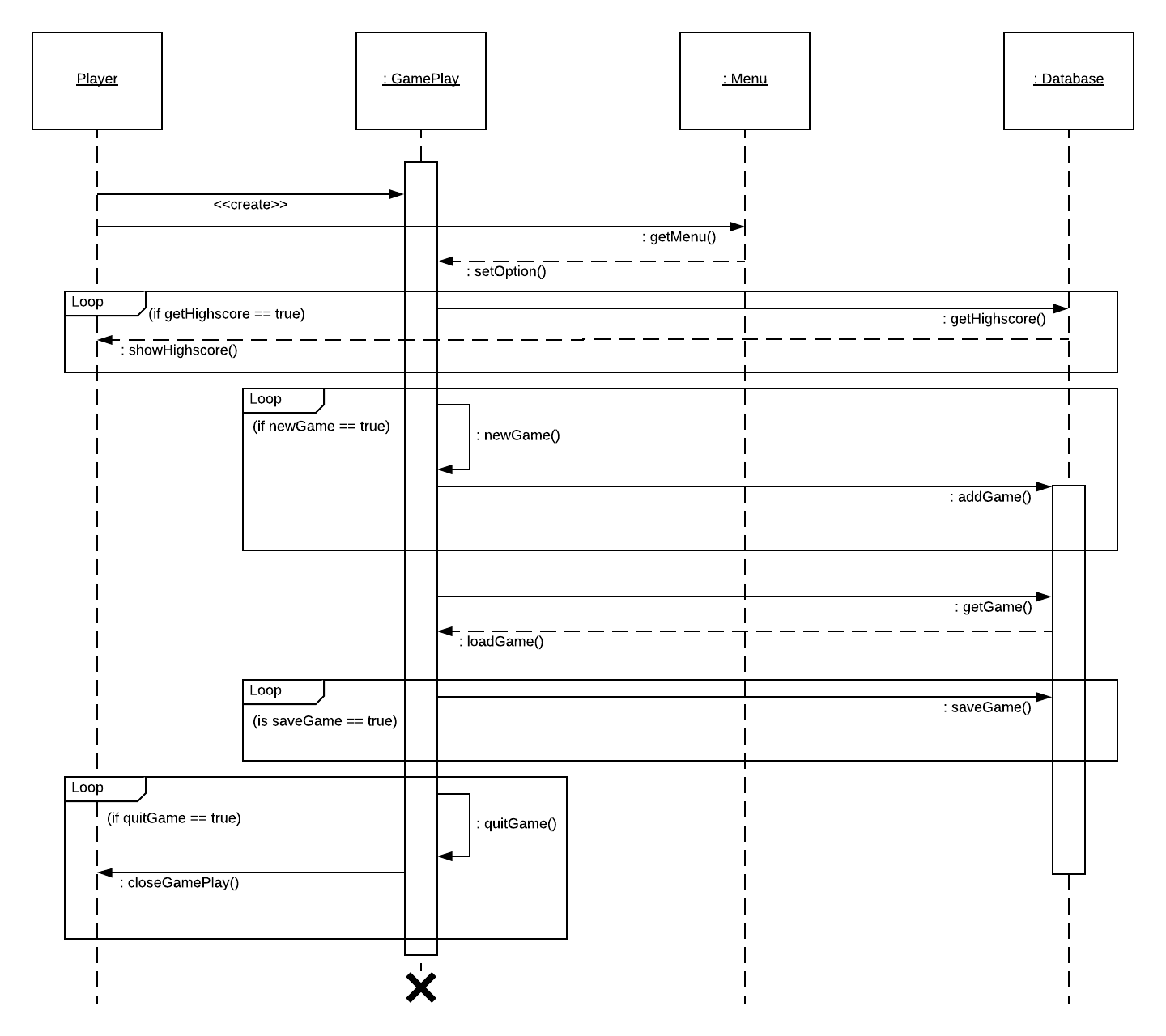


**Översikt av spelets huvudsakliga processer**

|  |  |
| --- | --- |
| **Process** | **Förklaring** |
| Controller | Den process som ansvarar för att ta emot input från användaren. |
| Pacman | Uppdaterar spelarens position på spelplanen samt kontrollerar om pacman kan äta spöken eller inte. |
| Ghost | Uppdaterar spökets position på spelplanen. |
| Menu | Den process som låter spelaren göra val att spara, ladda eller avsluta när spelet är pausat eller när användaren är i huvudmenyn. |
| Gameplay | Den process som ansvarar för att räkna poäng, upptäcka kollisioner och ha koll på om spelaren är i livet eller inte. Även den process som kommunicerar med databasen för sparade spel och hämtar dem om användaren begär det. |
| Render | Renderar spelets grafiska komponenter. |

2.4 Interactions

Följande diagram beskriver programmet från det att spelaren har skapat ett konto eller loggat in med befintligt konto. Spelaren kan sedan göra sitt val i menyn och diagrammet visar de funktioner som körs baserat på valet.



* Efter att GamePlay har skapats så kallas funktionen getMenu() som presenterar de olika valen som finns.
* När spelaren har gjort sitt val så skickas detta in i GamePlay där valet valideras och tilldelas med setOption().
* Sedan följer olika loopar/funktioner som körs beroende av spelarens val:  
  - getHighscore() hämtar data från databasen som sedan visas för spelaren.  
  - newGame() skapar ett nytt spel som sedan läggs till i databasen.  
  - loadGame() körs när ett sparat spel har hittats och hämtats från databasen.  
  - saveGame() sparar den aktuella sessionen, skriver över tidigare version i databasen.  
  - quitGame() avslutar GamePlay och förstör även det objektet.