Gruppnummer : 5 Deltagare: Eric och Jonathan Leverans : Pdf och github

# Projekt Fire dolphine

Gruppnummer : 5 Deltagare: Eric och Jonathan Leverans : Pdf och github

Gruppnummer : 5 Deltagare: Eric och Jonathan Leverans : Pdf och github

# 1 Inledning

I denna projekt ska göras en klassika skolans favorit hänga gubbe. Sådant spel vilken brukas på lekationen på t.ex. rasten eller lediga timmar.

I detta grupp 5 hade skapat ett kundens vision var att göra ett revulionärade spel av det och det ska inte vara som standard hänga gubbe. Gruppen heter Fire Dolphine för specifikt. Kunden har svårt veta vilka tekniska biståndsdelar behövs för konsutera denna projekt.

Inlämming av redovisingen av projektet hänvisar till datumet 28 januari år 2025. I denna projekt var två medlammar som ansvarde och lyckats bygga trotts brister i medlemerna. Namn på dessa medlemera heter Eric och Jonathan. För att hitta projektet kan obsaverad i denna github länk. (https://github.com/JonathanJirefalk/FireDolphinsHangman)

# 2 Reflektion

Under tidens som projektet var i verktsällade form fanns intresanta tilhörigheter som behövs relfelteras. För att läsa tydligare information i denna projekt hur allting startas kan obsveras i vid avsnitt ***2.3 Uppstart av Projektet.*** Samt hur hantering av mötet ansvarades och verkställdes kan läsas på ***avsnitt 2.4 Möteshantering. F***ör titta på kodexmplet finns avsnitt att obsveras i ***2.5 Kodning.*** Bland annat finns avsnitt att obsaveras i ***2.3.1 Min roll i implementationen och hur genomfördes*** som beskriver hur koden gemonfördes och vilken roll hade personen i projektet.

## 2.2 Uppstart av Projektet

Begynsle tid av projekt började med att studenten Eric kontakta alla på discord för leta om något som ville stödja och sammarbete i projektet. I uppdraget stod max fem personer i projektet. Därför blev bara en person kontaktat vilken var Jonathan. Vilken vi diskuterade i hur allt skulle uppbyggnas. För mer information hur mötet gick läs i avsnitt ***avsnitt 2.4 Möteshantering*** där vi beskriver hur första och sista mötet gick i processen.

## 2.2.1 Min roll i implementationen och hur genomfördes

**Min roll i implementationen**

Eric roll idenna projekt var lite som ta hand om och ansvara möten och verkställa projektet. Följe kunden önskan att bygga spelet och deras vison som stod spel. Samt även var kodare.

**Hur genomfördes**

Det började med Eric fick uppdrag att kontakta alla som skulle vara med gruppen Firedolphine. Vilken leder till bara en elev svarade och vilken var vid namnet Jonathan.

## 2.3 Möteshantering

Första mötet:

Eric inledde det första mötet, som ägde rum på ett datum som vi senare skulle fastställa. Mötet hade en brainstorming-karaktär tillsammans med Jonathan, där vi diskuterade hur spelet skulle se ut rent gränssnittsmässigt. Under denna konsultation beskrev vi även vilka programmeringsspråk vi skulle använda och beslutade att ta upp detta på det andra mötet.

Andra mötet:

Vid det andra mötet började gruppen diskutera att ha två medlemmar på grund av bristande kommunikation. Eric och Jonathan bestämde att de skulle driva projektet vidare och kom överens om att programmeringsspråket skulle vara Java, vilket var det språk de hade lärt sig under Java-kursen. Jonathan skapade även ett GitHub-projekt som vi laddade ner på våra datorer för att starta upp projektet.

Under tiden vi diskuterade på Discord programmerade vi samtidigt. Bland annat skapade Eric sin kod för "Välj ord-systemet" i Java, medan Jonathan ansvarade för de flesta koddelarna, såsom huvudfunktionerna i spelet.

Tredje mötet:

Det tredje mötet var planerat till 2025-01-24, men Eric fick tid att koda en layout till spelet och skrev på mötesdokumentet.

Fjärde mötet:

Det fjärde mötet, som ägde rum den 2025-01-26, blev inställt, men Eric arbetade på mötesdokumentet och Trello för att se vad som behövde göras.

Femte mötet:

Vid det femte mötet, som hölls den 2025-01-26, diskuterade jag och Jonathan spelet och vad som behövde göras. Jonathan granskade Erics kod för att se vad som behövde fixas i layouten.

Sjätte mötet:

Det sista mötet, som startade den 2025-01-27, innebar att Eric gick igenom GitHub och gjorde en pull request på koden. Han kontrollerade även alla fel som fanns i koden och kommenterade dem. Vi i gruppen tycker att Jonathan gör ett utmärkt jobb med kodningen. Eric ansvarade för alla möten och prioriterade kundkraven.

Mötesdokomänten länk:*https://github.com/JonathanJirefalk/FireDolphinsHangman/blob/main/M%C3%B6tesdokoment.pdf*

## 2.4 Kodning

```java
import java.util.Scanner;

public class Choosingwords {

    Scanner scanner = new Scanner(System.in);

    Layout layout = new Layout();

    public static String word1;
    public static String word2;

    public void chooseWord(){

        System.out.println(StartGame.playerOne + ", enter a word");
        word1 = scanner.nextLine().toLowerCase().trim();

        System.out.println();

        System.out.println(StartGame.playerTwo + ", enter a word");
        word2 = scanner.nextLine().toLowerCase().trim();

        System.out.println();

        layout.initializeGuesses(word1, word2);
        layout.displayGuessedWord(layout.getGuessedWord1(), StartGame.playerOne);
        layout.displayGuessedWord(layout.getGuessedWord2(), StartGame.playerTwo);
    }
}
```

Gruppnummer : 5 Deltagare: Eric och Jonathan Leverans : Pdf och github

På denna kod har Eric skrivet ett system i spelen där man välja ord. Kan besrkiva med psodo kod hur den fungerar

Klass Choosingwords

Skapa en Scanner för användarinmatning

Skapa en instans av Layout

Definiera strängar word1 och word2

Metod chooseWord()

Skriv ut meddelande för spelare ett att ange ett ord

Läs in ordet från spelare ett och spara det i word1

Konvertera word1 till gemener och ta bort eventuella ledande eller avslutande mellanslag

Skriv ut en tom rad

Skriv ut meddelande för spelare två att ange ett ord

Läs in ordet från spelare två och spara det i word2

Konvertera word2 till gemener och ta bort eventuella ledande eller avslutande mellanslag

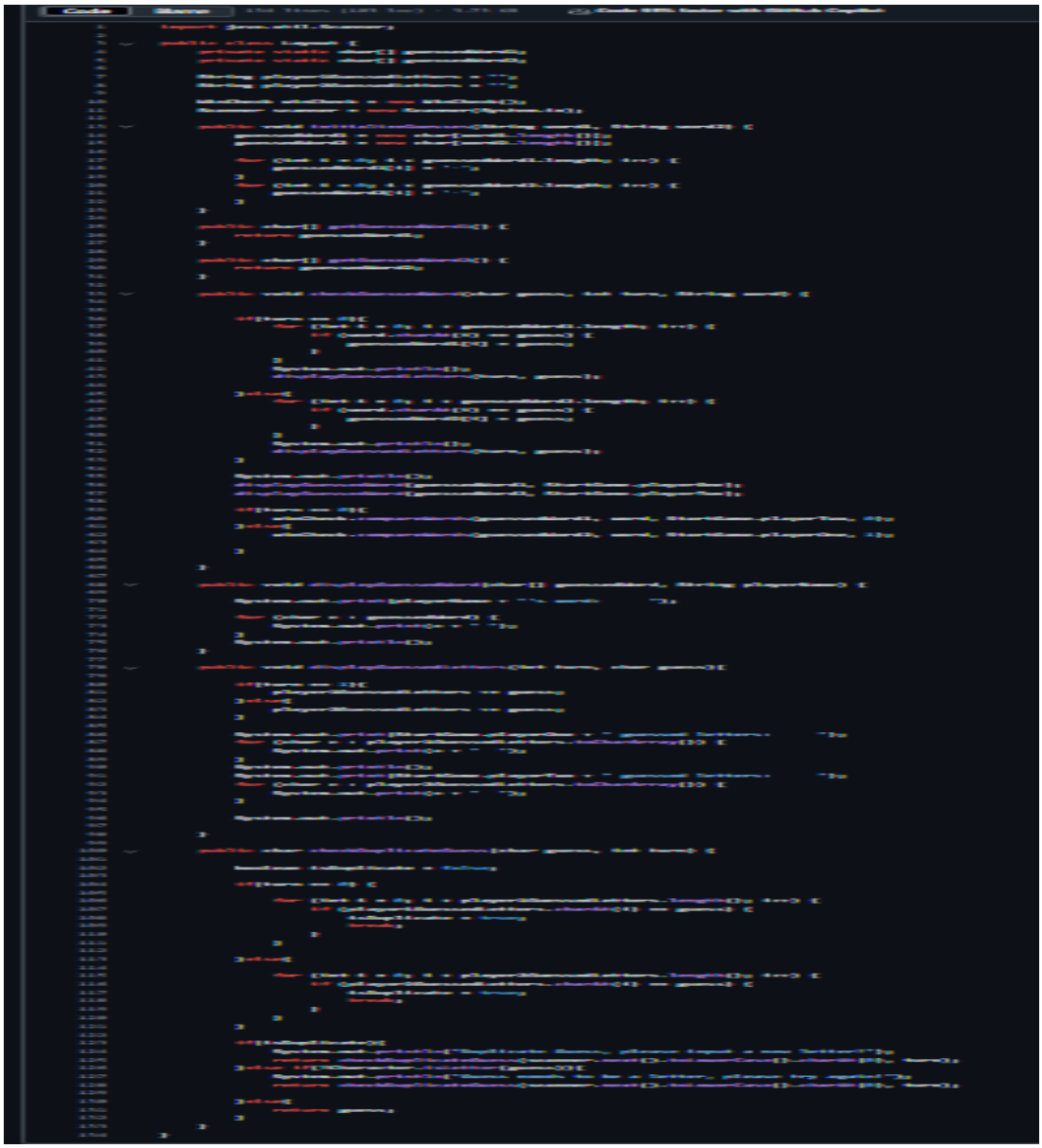
Skriv ut en tom rad

Anropa layout.initializeGuesses med word1 och word2

Anropa layout.displayGuessedWord med layout.getGuessedWord1() och StartGame.playerOne

Anropa layout.displayGuessedWord med layout.getGuessedWord2() och StartGame.playerTwo

Gruppnummer : 5 Deltagare: Eric och Jonathan Leverans : Pdf och github



Denna bild är extrem ut zoomad men förklarar koden psodokod för att förstå principen i koden.

Klass Layout

Definiera privata statiska teckenarrayer guessedWord1 och guessedWord2 för att lagra gissade bokstäver för varje spelare.

Definiera strängar player1GuessedLetters och player2GuessedLetters för att lagra gissade bokstäver av spelarna.

Skapa en instans av WinCheck för att kontrollera vinstvillkor.

Gruppnummer : 5 Deltagare: Eric och Jonathan Leverans : Pdf och github

Skapa en Scanner för användarinmatning.

Metod initializeGuesses(String word1, String word2)

Skapa arrayer guessedWord1 och guessedWord2 med längden av ord1 och ord2.

Fyll guessedWord1 med '-' för att representera ogissade bokstäver.

Fyll guessedWord2 med '-' för att representera ogissade bokstäver.

Metod getGuessedWord1()

Returnera guessedWord1.

Metod getGuessedWord2()

Returnera guessedWord2.

Metod checkGuessedWord(char guess, int turn, String word)

Om turen är 0 (spelare ett):

Gå igenom guessedWord1 och om gissningen matchar en bokstav i word, uppdatera guessedWord1 med gissningen.

Skriv ut en tom rad.

Anropa displayGuessedLetters för att visa gissade bokstäver för spelare ett.

Annars (spelare två):

Gå igenom guessedWord2 och om gissningen matchar en bokstav i word, uppdatera guessedWord2 med gissningen.

Skriv ut en tom rad.

Anropa displayGuessedLetters för att visa gissade bokstäver för spelare två.

Skriv ut en tom rad.

Anropa displayGuessedWord för att visa gissade ord för båda spelarna.

Om turen är 0, anropa winCheck.compareWords för att kontrollera om spelare två har vunnit.

Annars, anropa winCheck.compareWords för att kontrollera om spelare ett har vunnit.

Gruppnummer : 5 Deltagare: Eric och Jonathan Leverans : Pdf och github

Metod displayGuessedWord(char[] guessedWord, String playerName)

Skriv ut spelarens namn följt av deras gissade ord.

Gå igenom guessedWord och skriv ut varje bokstav.

Metod displayGuessedLetters(int turn, char guess)

Om turen är 1, lägg till gissningen till player1GuessedLetters.

Annars, lägg till gissningen till player2GuessedLetters.

Skriv ut gissade bokstäver för spelare ett.

Skriv ut gissade bokstäver för spelare två.

Metod checkDuplicateGuess(char guess, int turn)

Definiera en boolean isDuplicate som initialiseras till false.

Om turen är 0 (spelare ett):

Gå igenom player1GuessedLetters och kontrollera om gissningen redan har gjorts.

Om så är fallet, sätt isDuplicate till true.

Annars (spelare två):

Gå igenom player2GuessedLetters och kontrollera om gissningen redan har gjorts.

Om så är fallet, sätt isDuplicate till true.

Om isDuplicate är true:

Skriv ut ett meddelande om att gissningen är en duplikat och be om en ny bokstav.

Anropa checkDuplicateGuess igen med den nya gissningen.

Annars, om gissningen inte är en bokstav:

Skriv ut ett meddelande om att gissningen måste vara en bokstav och be om en ny bokstav.

Anropa checkDuplicateGuess igen med den nya gissningen.

Annars:

Returnera gissningen.

Ifall på denna kod på layout har då jonthan gjord ändringar till mer rätt.

# 5 Samarbete och versionshantering

- Vilka branch-strategier använde ni?

Den stragerie som vi anvädnes oss på t.ex. vi har en main branch medans vi har två egna branch som vi laddar upp varsin delar och sen merge ihop. Så vi har används av oss main metoden snabb och enkel på grund av kort vecka och arbetstid.

- Hur hanterade ni merge conflicts?

Vi löste genom att kontroller koden och löste med manuellt konrtollera filen som vår editor tas upp.

- Vilka Git-kommandon var mest användbara?

**git remote add origin**

**git add**

**git commit –m**

**git push origin main**

- Hur strukturerade ni ert GitHub-repo?

**FireDolphinsHangman**/src

På src så finns alla kod som vi skrev in och ladda upp github.

Exempel på problem som stötte på kan vara .tex. Visa typ av kod fungerer inte med varandra vilken behövdes fin puts. Som t.ex. på kod layout var en kort kod i från början men blev längre på grund av saker som inte fungerade i systemet.

# 6 Avslutning

I vårt projekt hade vi en tydlig och strukturerad metod för att hantera backloggen, vilket var avgörande för vår framgång. Vi inledde med att definiera och prioritera uppgifter i Trello, vilket gjorde det möjligt för oss att visualisera arbetsflödet och identifiera de mest kritiska uppgifterna. Eric tog ansvar för att leda mötena och dokumentera diskussioner från möten. vilket i gruppen säkerställde att alla var på samma sida av beslutet.

Jonathan fokuserade på kodningen och den tekniska implementationen. Denna uppdelning av ansvar gjorde att vi kunde arbeta effektivt och utnyttja våra styrkor.

Vi tog ett solidariskt ansvar för projektet, vilket innebar att vi stöttade varandra och hjälpte till där det behövdes. Om någon stötte på hinder var vi snabba att erbjuda hjälp och diskutera lösningar tillsammans.

Gruppen hanterade tanken på code reviews på ett bra sätt, även om vi bara hade en person att arbeta med. En av de mest komplexa och tidskrävande aspekterna var att säkerställa att varje fil integrerades korrekt med varandra. Det var avgörande att koden inte hamnade i konflikt med andra delar av projektet. För att förbättra denna process skulle vi kunna införa fler regelbundna code reviews och automatiserade tester för att tidigt upptäcka och åtgärda potentiella konflikter.