

# Inledning

Dimitri Mostachjov.

2022-12-23

## Kundens vision:

Innan utvecklingen av spelet startade, mottogs muntliga och skriftliga förfrågningar från klienten. Enligt dessa önskemål såg klientens vision av spelet ut så att ingen av spelarna behövde vänta på sin tur under spelets gång. Det var nödvändigt att implementera klientens idé så att flera spelare kunde delta i spelet samtidigt. Som ett inledande projekt, övervägdes alternativet när det skulle finnas två spelscener på skärmen samtidigt. Således kommer två spelare att kunna gissa ord samtidigt. Det fanns även en förväntad önskan att det vore bra om det fanns möjlighet att spela fyra spelare samtidigt. Beställaren ser vidareutvecklingen av projektet i att det tillkommer möjligheter att spela i lag, till exempel två spelare mot två andra.

## Gruppens github repo:

[https://github.com/MasadanThe/Revolutionary\\_Hangman.git](https://github.com/MasadanThe/Revolutionary_Hangman.git)

# Reflektion

## User stories.

Här kommer jag att beskriva tre **user stories** som jag deltagit i utvecklingen av.

### 1. Förbjud användningen av en slumpmässig vokal för ett givet antal gånger.

En av kundens önskemål var att kunna inaktivera valet av en slumpmässig vokal under ett visst antal gånger. Direkt implementerad av Oskar.

```
public char randomizeForbiddenVowel() {  
    Random random = new Random();  
    char[] forbiddenVowels = {'a', 'e', 'i', 'o', 'u', 'y', 'å', 'ä', 'ö'};  
    // Randomize which index of the character that will be forbidden.  
    int forbiddenVowelIndex = random.nextInt( bound: forbiddenVowels.length - 1);  
    System.out.println("Banned vowel is: " + forbiddenVowels[forbiddenVowelIndex]);  
    return forbiddenVowels[forbiddenVowelIndex];  
}
```

*Dimitri arbetade med nästkommande del av detta storie:*

Formulering av user storie i Trello: Inställning på hur många rundor en bokstav ska vara förbjuden, ska göras knappar i **settingsScene**. Valet ska sparas i **Int disabledRounds** i Hangman.java

Först skrev jag kod som skapar fysiska radioknappar i den visuella delen av programmet. Här ger jag ett exempel på en del av koden.

```

RadioButton buttonSetDisabledRounds1 = new RadioButton(s: "Random vowel will be banned for 1 round");
RadioButton buttonSetDisabledRounds2 = new RadioButton(s: "Random vowel will be banned for 2 rounds");
RadioButton buttonSetDisabledRounds3 = new RadioButton(s: "Random vowel will be banned for 3 rounds");
buttonSetDisabledRounds1.setLayoutX(550);
buttonSetDisabledRounds1.setLayoutY(110);
buttonSetDisabledRounds2.setLayoutX(550);
buttonSetDisabledRounds2.setLayoutY(130);
buttonSetDisabledRounds3.setLayoutX(550);
buttonSetDisabledRounds3.setLayoutY(150);
buttonSetDisabledRounds1.setOnAction(e -> disabledRounds = 0);
buttonSetDisabledRounds2.setOnAction(e -> disabledRounds = 1);
buttonSetDisabledRounds3.setOnAction(e -> disabledRounds = 2);

ToggleGroup bannedVowelBut = new ToggleGroup();
buttonSetDisabledRounds1.setToggleGroup(bannedVowelBut);
buttonSetDisabledRounds2.setToggleGroup(bannedVowelBut);
buttonSetDisabledRounds3.setToggleGroup(bannedVowelBut);

settingsGroup.getChildren().add(buttonSetDisabledRounds1);
settingsGroup.getChildren().add(buttonSetDisabledRounds2);
settingsGroup.getChildren().add(buttonSetDisabledRounds3);

```

Metoden **randomizeForbiddenVowel** anropas i **createChooseWordGroup** -metoden när man anger ord som kommer att vara en gissningsuppgift.

```

2 usages pemillamoreno +2
public Group createChooseWordGroup() {

    Group chooseWordGroup = new Group();
    TextField chooseWord = new TextField();
    chooseWord.setLayoutX(550);
    chooseWord.setLayoutY(90);

    chooseWordGroup.getChildren().add(chooseWord);

    Button enterWordButton = new Button(s: "Enter your word");
    enterWordButton.setLayoutX(550);
    enterWordButton.setLayoutY(130);
    enterWordButton.setOnAction(actionEvent -> {
        String word = chooseWord.getText();
        Player player = playersList.get(playerIndex);
        player.setWord(word);
        player.updateText();

        if (playerIndex == playersList.size() - 1) {
            changeToPlay();
        }
        chooseWord.clear();
        playerIndex++; // next player be able to enter word
        randomizeForbiddenVowel();
    });
    chooseWordGroup.getChildren().add(enterWordButton);

    return chooseWordGroup;
}

```

Jag lade också till kod i **checkForCharacter** -metoden för att räkna antalet omgångar när en förbjuden vokal inte kommer att anges. När det inställda antalet omgångar slutar låses vokalen upp.

```

public boolean checkForCharacter(char character) {

    boolean foundChar = false;
    if (disabledRoundsLetter > 0 && character == Hangman.getForbiddenVowel()) // Dimitri
    {
    } else {
        // Checks if the Character in word
        for (int i = 0; i < word.length(); i++) {

            if (word.charAt(i) == character) {
                foundChar = true;
                wordAsArrayList.set(i, character);
                System.out.println(wordAsArrayList.toString());
            }
        }

        if(!foundChar)
        {
            //Adds the character to the list
            guessedWrong.add(character);
        }

        if (disabledRoundsLetter > 0){ // Dimitri
            disabledRoundsLetter--;
        }
    }

    return foundChar;
}

```

## 2. Spelaren ville att när en felaktig bokstav skrevs in, dök delar av galgen och figurer upp på skärmen.

Formulering av user storie i Trello: I Player.java skapa metoden **wrongWord** som loopar igenom drawing i man och sätter nästa del i teckningen till isVisible(true)

För att få den här funktionen att fungera skrev jag och implementerade metoden "wrongWord".

```

1 usage  ⤴ Dimitri
public void wrongWord () {
    man.getDrawing().get(wrongLetterCounter).setVisible(true);
    wrongLetterCounter++;
}

```

Metoden wrongWord anropas i **Play** -metoden om spelaren anger en felaktig bokstav.

```

//Look for the player who gets guessed on and check if it is not the right character
if (!playersList.get(order.get(turn)).checkForCharacter(pressedKey))
{
    //Sets the player that guessed wrong to start getting hanged
    playersList.get(turn).wrongWord();
}

```

Under utvecklingen av programmet var elementen i spelscenen (galgen) först synliga för att underlätta arbetet med utvecklingen av programmet, när programmet nästan var klart satte jag läget för elementen till osynligt. Till exempel: `base.setVisible(false)`. Element ändrar läge till `setVisible(true)` när spelaren skriver in fel bokstav.

### 3. Övergång från `selectWordScene` till `playScene` efter att alla spelare har angett ett gissningsord.

Andemeningen med denna metod är att ändra scenen för inmatning av ord till scenen för själva spelet. Metoden `changeToPlay` implementerad av mig.

```
public void changeToPlay() {  
  
    stage.setScene(playScene);  
  
}
```

För att anropa denna metod lades följande kod till i `setOnAction` -metoden för `enterWordButton` -knappen:

```
if (playerIndex == playersList.size() - 1) {  
  
    changeToPlay();  
  
}
```

## Avslutning

Under arbetet med projektet visade medlemmarna i vår grupp, som var kvar i arbetet med projektet: Pernilla, Oskar, Dimitri en god samarbetsförmåga. Vi har alltid varit öppna för kontakter med varandra. Vi skapade en grupp i Discord, ett projektkonto i Trello. Vi höll diskussioner om lösningar i Discord, efter fördelningen av individuella uppgifter registrerade vi in det i Trello. Nästan dagligen hade vi röstmöten i Discord, för korrespondens i Discord var vi alltid tillgängliga på arbetstid och lite bredare än arbetstid.

**Jag betraktar min roll** i projektet som rollen som en projektdeltagare som var intresserad av och deltog i olika typer av programmeringsarbete. Det hjälper mig att få kläm på olika typer av programmeringsarbete.

**Vi fyllde på backlog** med information efter det första mötet med kunden. Under arbete kompletterade vi informationen i backlog. Vi flyttade uppgifter från backlog till avsnittet "Developing", ibland återförde vi uppgifter till backlog om uppgifterna orsakade svårigheter eller ytterligare frågor uppstod och det var nödvändigt att ägna ytterligare uppmärksamhet åt dessa uppgifter. Alla medlemmar i teamet kunde ändra informationen i backlog-sektionen.

**Vid fördelning av uppgifter** mellan gruppmedlemmarna fanns en princip om frivilligt val, samt beredskap att börja arbeta med vilken uppgift som behövs som var aktuell vid den tiden.

Vissa uppgifter, särskilt i början av arbetet med projektet, fördelades av den informella gruppleddaren Oskar Nylund. När arbetet med projektet tog fart, ungefär fem dagar efter projektets start, började gruppmedlemmarna frivilligt och på eget initiativ föreslå att ta ansvar på sig för någon del av projektet.

**Det fanns ingen kritik** i bokstavlig mening av ordet "kritik" i vår grupp, enligt min mening. Jag skulle använda definitionen av "vänliga rekommendationer från kollegor." Detta tillvägagångssätt påverkade oss mycket positivt. Vi har alltid varit redo att ta del av kollegornas uppgifter för att påverka utvecklingen av vårt projekt. I princip är detta tillvägagångssätt fördelaktigt för alla, inte bara för att det är nödvändigt att leda själva projektet framåt, utan också för att med detta tillvägagångssätt går varje deltagare igenom professionell utveckling genom erfarenhetsutbyte.

**De svårhanterade problemen** i vårt projekt var främst av typen av arbetsplanering mellan teammedlemmar. Detta är endast relevant i början av arbetet med projektet. Sedan försvann den här typen av problem. Kärnan i problemet är att fem deltagare tilldelades projektet. En av deltagarna kom aldrig till mötet. Den andra deltagaren i projektet lämnade projektet efter några dagar. Under projektets början kunde vi inte planera fördelningen av arbetsomfattningen, eftersom det var en instabil situation med deltagarna. Vi visste inte exakt hur många som skulle göra projektet. Med tanke på att vi inte visste exakt vilken professionell nivå varje projektdeltagare hade vid den tiden, hade vi svårt att utveckla de tekniska aspekterna av projektgenomförandet. När sammansättningen av gruppmedlemmarna blev tydlig och stabil började projektet genomföras utan betydande svårigheter. Som "alla" vet: grunden för projektets framgång är ett team som passar varandra.

Rent tekniskt, i programmering, hade gruppen inga olösliga problem. En del av den kompetens vi fick av vår informell ledare, en del av den kunskap vi hämtade från externa källor: Internet.

**Under projektets gång använde vi ett "Agile" förhållningssätt.** Vi organiserade oss själv, vi var flexibla och öppna för att hitta lösningar i projektet, vi reagerade snabbt om det var nödvändigt att göra förändringar eller ta del av andra deltagares uppgifter.

Specifikt verkar det vara svårt för mig att avgöra vilken av metoderna som är typiska i Agile vi använde. Jag tror att Scrum skulle kunnat vara rätt för oss, men i början av projektet, förmodligen på grund av instabiliteten i teamet, tappade vi lite kontroll över processen. Efter att kunden visade sin missbelåtenhet på att vi inte hade en prototyp klar vid överenskommen tid, ändrade vi tillvägagångssätt. Jag tror att vårt tillvägagångssätt kan beskrivas som en kombination mellan Trunk Base Development och Kanban, men jag tror att jag kan säga att vi också hade inslag av Scrum-metoden, till exempel jag skulle kunnat säga att Oskar hade scrum-mästare roll i vårt projekt.

Såvitt jag förstår av rekommendationerna bör lag som precis har börjat använda Agile endast välja en metod: antingen Scrum eller Kanban. Det kan konstateras att vi i detta projekt inte lyckades att följa endast en av dessa metoder.

Detta är en av många andra professionella färdigheter som vi behöver utveckla.