

BeardDefender

Ramverk: Monogame

Genre: 2D PlatformSpel

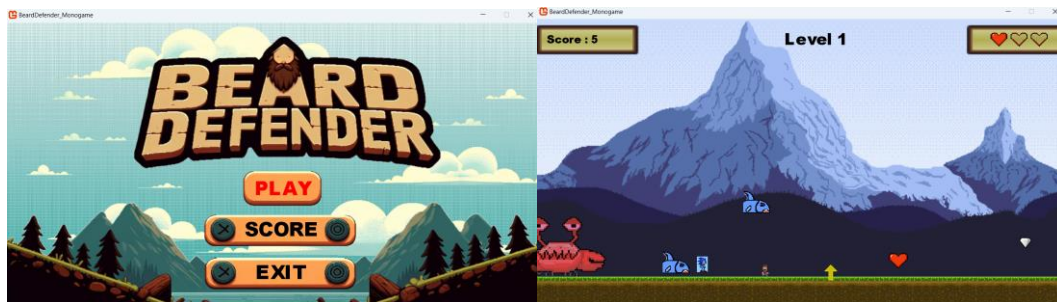
Utvecklingsteam: Rasmus Broman, Jonas Öhrn, Nikolaos Prevenios, Ahmed Dirie, Olle Ristenstrand

Kravspecifikation Exakt vilket spel och hur det skall se ut skall beskrivas i kravspecifikationen.

Kravspecifikationen skall innehålla en beskrivning av projektet i en flödande text (och allra helst med en eller flera bilder). Detta utgör er vision över det färdiga systemet. Därefter skall ett antal konkreta krav följa (i tabellform). Det är då viktigt att kraven är tydliga, verifierbara, numrerade och prioriterade. Kraven måste också minst täcka de minimikrav som vi har på projektarbetet. Kravspecifikationen skall skickas som en PDF-fil via Learnpoint. Kravspecen ska också läggas upp i erat github-projekt så att det blir lätt för er, och mig att gå tillbaka och kolla kraven.

Minimikrav - Funktionella

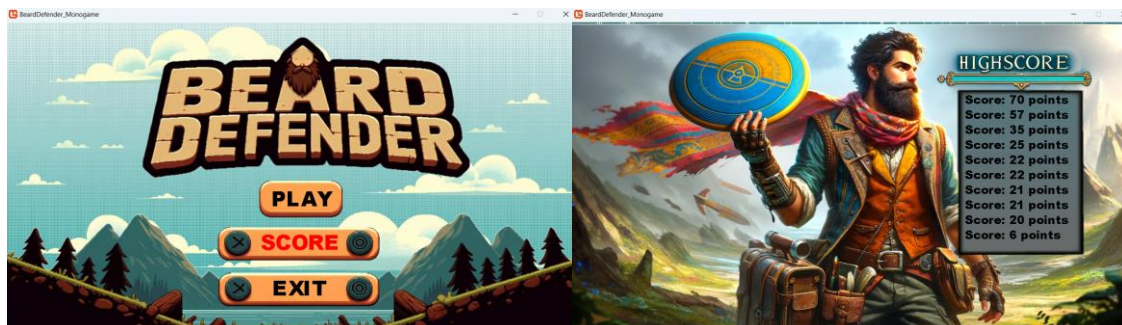
1•Projektet skall bestå av ett 1-spelarspel med grafiskt gränssnitt.



2•Gränssnittet skall styras med tangentbord och/eller mus. Det är inte krav på båda ska användas.

Styrs med piltangenterna, hoppa med space. Även WASD fungerar att använda

3•Spelet skall ha en "meny", med några alternativ (t.ex. "nytt spel", "highscore" och "avsluta").

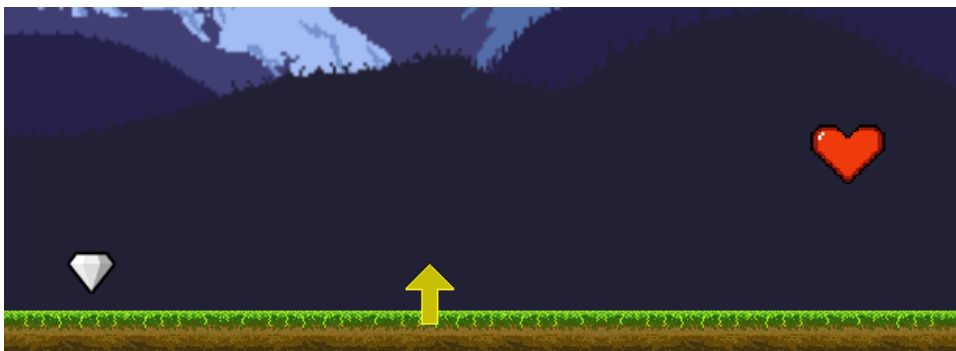


4•Spelet skall åtminstone ha två "banor", minst två olika typer av "fiender" och minst två olika typer av "power-ups". För att det ska räknas måste funktionaliteten skilja. Ex: Ett monster jagar spelaren, det andra rör sig slumpmässigt, en powerup ger poäng medan en annan ger extra liv. Det måste alltså skilja tillräckligt för att motivera att det ska vara två olika klasser.



Crabman som står redo att äta spelaren om man hamnar längst till vänster av skärmen.

Flying-shark som rör sig över himlen för att försvåra för spelaren.



Diamond - för mer poäng

Arrow – Jump boost

Heart – Mer hälsa

5•Spelet skall lagra poäng från tidigare spel i en "highscore" eller liknande. Detta måste lagras på datorns hårddisk. Minimikrav - Konceptuella



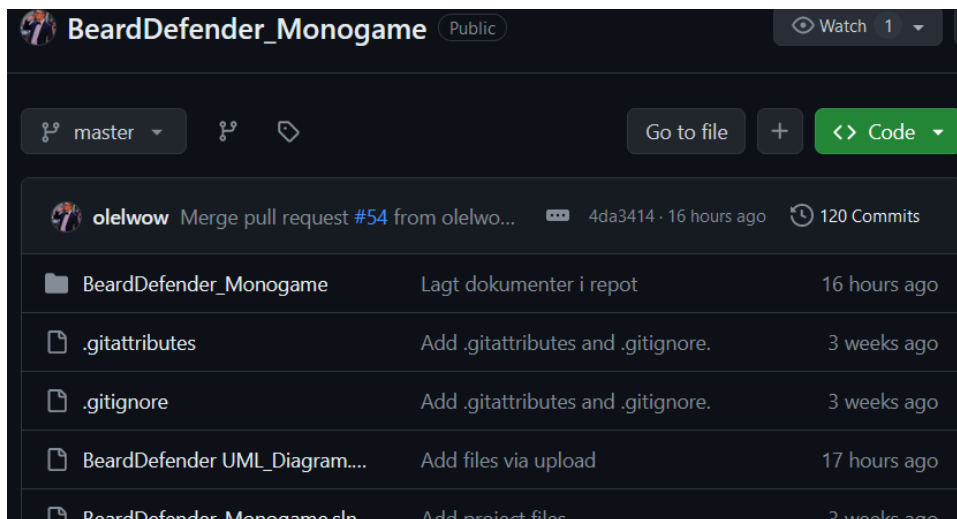
Fil som sparas på skrivbordet där vår "highscore" hamnar.

6•Projektet skall göras objektorienterat och vara skrivet i C#.

```
7 references
internal class Crabman : Enemy
{
    //RectangleF position = new RectangleF();
    private Texture2D currentTexture;
    private Texture2D[] textureArray; // Innehåller texturer för rörelse
    float frameDuration = 0.2f; // Hastighet på animationen.
    float frameTimer = 0f;
    int currentFrameIndex; // Variabel som väljer vilken frame man tar u
    private int positionX = 40;
    private int positionY = 586;

    1 reference
    public Crabman()
    {
        // Array MÅSTE intieras innan man försöker lägga in textures i d
        this.textureArray = new Texture2D[3];
    }
}
```

7•Projektet skall versionshanteras med Git.



The screenshot shows the GitHub interface for the repository "BeardDefender_Monogame". At the top, the repository name is displayed with a "Public" badge and a "Watch" button showing 1 watch. Below this, there are navigation buttons for "Go to file", a "+" button, and a "Code" button. The main section shows a list of commits. The most recent commit is by "olelwow" titled "Merge pull request #54 from olelwo...", dated "16 hours ago", with "4da3414" as the commit hash and "120 Commits" in the history. Below the commit list, there is a table of files in the repository:

File Name	Description	Time
BeardDefender_Monogame	Lagt dokumenter i repot	16 hours ago
.gitattributes	Add .gitattributes and .gitignore.	3 weeks ago
.gitignore	Add .gitattributes and .gitignore.	3 weeks ago
BeardDefender UML_Diagram....	Add files via upload	17 hours ago
BeardDefender_Monogame.sln	Add project files	3 weeks ago