

Projektbeskrivning

Project: AlphaSigmaSigma

2032-03-22

Projektmedlemmar:

Hugo Nilsson <hugni385@student.liu.se>
Christoffer Näs <chrna581@student.liu.se>

Handledare:

Jonathan Falk <jonfa001@student.liu.se>

Innehåll

1. Introduktion till projektet	2
2. Ytterligare bakgrundsinformation	2
3. Milstolpar	2
4. Övriga implementationsförberedelser	4
5. Utveckling och samarbete	4
6. Implementationsbeskrivning	6
6.1. Milstolpar	6
6.2. Dokumentation för programstruktur, med UML-diagram	6
7. Användarmanual	7

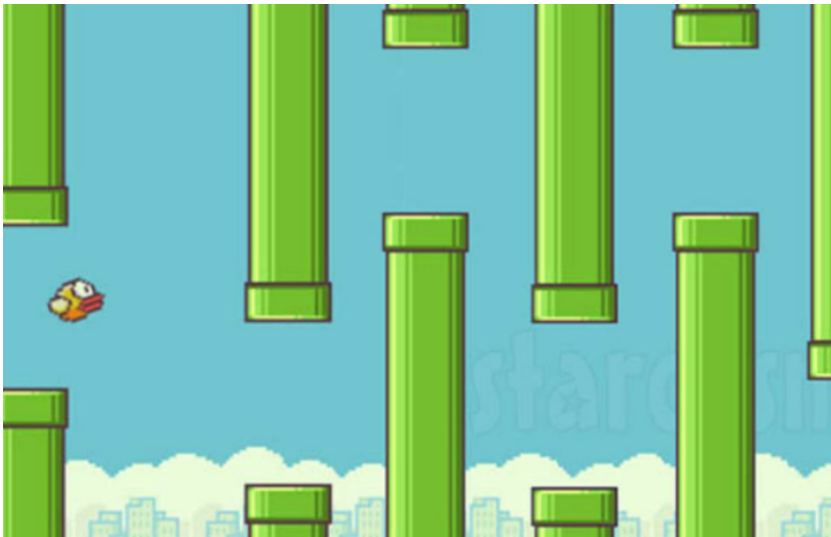
Projektplan

1. Introduktion till projektet

Vi tänker utveckla ett spel som liknar flappy bird, och vi har då utgått ifrån inspirationsprojekt av typen Vertically/Side Scrolling.

vilket innebär att spelarens karaktär rör sig framåt längs banan genom att spelplanen rör sig i motsatt riktning. Vårt mål med spelet är att samla så mycket poäng som möjligt. Spelet avslutas när man dör. Till spelarens hjälp kommer olika power-ups finnas tillgängliga.

Det slutförda projektet kommer likna något i denna stil, fast med sämre grafik då vi kommer att prioritera ett välfungerande spel över snyggt utseende:



2. Ytterligare bakgrundsinformation

Vårt spel kommer att vara av typen side-scroller vilket innebär att spelet följer spelaren allt eftersom den förflyttar sig höger eller vänster. Denna genre av spel var som mest populära tidigt i spelutvecklingens historia då hårdvarans begränsningar var större. För ytterligare läsning följ denna länk:

https://en.wikipedia.org/wiki/Side-scrolling_video_game

3. Milstolpar

#	Beskrivning
1	Spelplan/Fönster
2	Karaktären
3	Hitbox
4	Hinder
5	Kollision med hinder
6	Spelbart spel
7	Poäng
8	Bytbar Karaktär
9	Power-Ups-Storleksförändring
10	Power-Ups-Skjuta
11	Power-up extra liv
12	Highscore
13	Rörbara hinder
14	Svårighetsgrad

Spelplan: Fönstret som spelet spelas i.

Spelkaraktären: En karaktär som spelaren kan styra uppåt och nedåt. Spelplanen rör sig så att spelkaraktären är kvar centrerat i skärmen.

Hitbox: Spelkaraktären och Spelplanens översta och understa kant har nu någon slags fysisk representation, en hitbox.

Hinder: Hinder som spelar karaktären kan kollidera med. Hindren har både grafisk och fysisk representation.

Kollision: En Collisionhandler som hanterar kollision mellan spelkaraktären och hinder. Eller kollision emellan spelaren och Spelplanens botten eller toppen.

Spelbart: Ett spelbart spel där man kan i någon utsträckning spela spelet och man kan förlora genom att kollidera.

Poäng: Lägg till någon slags poäng räkning som, poängen ges genom att samla coins.

Bytbar Karaktär: Möjlighet att kunna byta spelkaraktär.

Power Ups: Power Ups är temporära eller permanenta förbättringar som spelaren kan samla på sig. Exempel på det här den röda svampen, fjädern och stjärnan i Super Mario spelen.

Highscore: Som nämnt ovan kommer spelaren kunna samla objekt som i sin tur ger poäng. Dessa poäng kommer sparas och jämföras, varpå en highscore-lista kommer att vara tillgänglig för spelaren på startmenyn.

Kollision: En stor del av spelets funktionalitet bygger på att de olika objekten faktiskt kan kollidera och således interagera på olika sätt. Därför kommer vi att implementera någon form av fysikmotor som kan hantera alla olika typer av kollisioner.

4. Övriga implementationsförberedelser

Man kan ibland spara mycket tid på att först ta en ordentlig funderare på t.ex. vad programmet ska göra, ungefär hur funktionaliteten kan delas upp mellan olika klasser, osv.

Vi tänkte att vårt spel skulle kunna byggas upp utav dessa olika klasser, fler kan självklart behövs skapas under utvecklingens gång.

Klass, Spelplan

Klass, Spelplan viewer, Ritar upp en spelplan i ett fönster så spelet kan spelas.

Klass, Spelkaraktär - Beskriver spelkaraktärens, storlek, färg, har fysisk och grafisk representation

Klass, Spelkaraktär maker, skapar en Spelkaraktär

Klass Collisionhandler, hanterar kollision emellan spelkaraktären och andra objekt.

Klass, EpicBirdComponent, används för att rita upp grafik.

Klass, Hinder, har fysisk och grafisk representation

Klass, Hinder maker, skapar upp ett hinder.

Klass, Powerups, har fysisk och grafisk representation

Klass, Powerups maker, skapar upp en powerup.

5. Utveckling och samarbete

Vi planerar på att arbeta tillsammans en stor del av projektet, för att sedan när det går ta på oss egna uppgifter såsom att exempelvis implementera olika powerups. Vi kommer att arbeta med projektet under gemensamma överenskomna tider.

Projektet kommer att ta sin början efter tenta-perioden under VT1 då vi i gruppen har mycket annat som behöver fokus och tid.

Tidsplan:

1 / 4 - 2022 - Vi har påbörjat projektet, Startat upp git repon
- Skapa ett öppningsbart fönster

1 / 5 - 2022 - Spelbart spel Mål 6. Utan features

17/5 -Färdigt spel.
- Felsökning och polishing.

22 / 5 - Hård deadline
- Demonstrera och lämna in

Projektrapport

6. Implementationsbeskrivning

6.1. Milstolpar

6.2. Dokumentation för programstruktur, med UML-diagram

7. Användarmanual