**Bebisspelet**

* Namn: Lavin Bamarni
* Klass: 200S
* Datum: 2021-05-14

Innehåll

[**Spel Projektet** 1](#_Toc71845938)

[**Projektbeskrivning, tekniker och planeringen** 2](#_Toc71845939)

[**Problem & lösningar** 3](#_Toc71845940)

[**Resultat & avslutning** 4](#_Toc71845941)

### **Spel Projektet**

I det här projektet har byggdes ett spel med Unity, spelet namngavs till Bebisspelet och handlar om en bebis som undviker olika hinder för att överleva. Några exempel på dessa hinder är ”fireballs”, svarta katter, yxor och *coola* ankor. För att kunna röra på bebisen trycker man endast på en knapp, uppåt pilen på keyboarden. Bebisen ska från början kunna röra sig framåt automatiskt och när du trycker på uppåt pilen rör den sig uppåt, på detta sett ska du försöka få bebisen i säkerhet.

#### Inspiration

I början var tanken annorlunda om hur spelet skulle vara. I slutändan blev spelet bättre än förväntat. Ursprungligen kom idén om spelet utifrån inspiration utav andra spel. Några av dessa spel är till exempel; Geometry Dash och Flappy Bird. Det som dom två spelen har gemensamt är att huvudkaraktären kan flyga i båda och att man undviker hinder. Inspirationen som spelet Flappy Bird utgav var att huvudkaraktären alltid skulle flyga, det negativa med Flappy Bird var att spelet aldrig tog slut, man kom aldrig till slutet av spelet då det inte fanns ett. Därför togs inspiration av Geometry Dash när ett ett slut lades till i spelet. I både Geometry Dash och Flappy Bird finns det en *restart* och en *play* *again* knapp som gjorde att spelet började om och det var omtyckt så därför programmerades det in i bebisspelet också.

### **Projektbeskrivning, tekniker och planeringen**

Unity är en spelmotor där man kan utveckla diverse applikationer med olika funktioner. Med Unity kan man skapa 3D, 2D, VR & AR spel, filmer och mer. Bebisspelet är byggt i Unitys 2D funktion.

**Graphical user interface, application

Description automatically generated**För att lägga till olika rörelser på några GameObject användes många olika Components. Vissa av dom är till exempel: *RigidBody*. När en kod skulle läggas till skapades en *New Script* komponent, där koden skrevs. För att några GameObjects skulle kunna kollidera med varandra användes *BoxCollider2D*.   
Jag använde mig utav Planner för att strukturera och bryta ner arbetet och få en översikt på vad som måste göras. I plannern hade jag tre buckets; Planering, Göra spelet och Utvärdering. I *Planering* hade jag skrivit ner vad jag behövde förbereda inför projektet. Där skulle jag till exempel hitta mina GameObjects eller olika koder som jag skulle ha användning av i spelet. I *Göra Spelet* var självaste skapandet av spelet i focus. Där skrevs det ner till exempel *”Koppla Github till Unity”*. I sista bucketen, *Utvärdering,* fokuserar man mest på att få ett avslut på hela projektet. Därför finns det till exempel punkter som *”Få feedback”* och *”Ge Feedback”*.

### **Problem & lösningar**

Under tiden som spelet skapades uppkom några problem, som gick att lösa i slutändan. Det problemet som tog längst tid att lösa hade med *fireball* GameObjects att göra. Det svåraste var att skriva koden och få den att fungera då det alltid uppkom errors. Om problemet inte var errors fanns det ett annat och det var att *fireballs* inte bara gick från höger till vänster utan dom kom tillbaka till startpositionen då två punkter användes, punkt A och punkt B. Detta fixades genom att en del av koden kommenterades bort. I stället för att använda den koden som kommenterades bort till *fireballs* användes den på en annan GameObject som kallas ”*Axe”* som också var ett hinder i spelet.

Sedan var detta projekt en ganska stor utmaning för mig då jag själv valde att komma på ett eget spel som ingen hade skapat tidigare och därför fanns det inga specifika tutorials, utan jag fick alltid testa mig fram.

Det som var enklast att göra var scenerna och att få knapparna att funka mellan scenerna. Som att till exempel skapa en Main Menu scen och från den scenen kunna gå till en annan scen genom att klicka på en knapp. Text

Description automatically generated

## **Resultat & avslutning**

Med alla problem och lösningar som uppstod lyckades jag ändå bli klar med spelet i tid och resultatet blev bättre än förväntat, då jag inte tidigare haft några erfarenheter av att skapa spel. Om jag hade haft mer tid på mig skulle jag lägga till mer nivåer i spelet. Jag skulle även öka svårighetsgraden på varje nivå som skapas. Grundtanken med spelets nivåer var att alla skulle vara riktigt svåra och jag lyckades få första nivån svår till en viss grad. Detta bevisades när jag lät två personerna testa mitt spel. Jag såg att båda ville klara nivån även när den var riktigt svår. I början var det svårt men efter många försök klarade dom nivån. Jag fick även feedback på spelet av dom:

1. Rebecca (Klasskamrat/Vän):

Gillade att spelet börjades automatiskt när man klickade på ”*start playing*” knappen då det var en utmaning att inte förlora från början. Hon tyckte även att det var roligt, att man ville fortsätta spela då det var svårt att klara av nivån. Sedan gillade hon att man hade tydliga instruktioner som förklarar spelet utan att behöva fråga skaparen av spelet hur man spelar.

Det hon tyckte jag borde lägga till i spelet var mer nivåer som hade större utmaningar.

1. Davin (Syster):

Hon tyckte om svårighetsgraden av spelet då det fick henne att fortsätta vilja spela.

Det hon tycker att jag borde lägga till var några skins som man skulle kunna välja mellan. Hon ville även att det skulle finnas en funktion som räknar hur många gånger man kört innan man vunnit, så varje gång spelaren förlorade skulle det finnas en funktion som visades någonstans på skärmen hur många gånger som spelaren kört.