

Agil projektledning

Utveckling av indie spel

Översikt

Projektet syftar till att utveckla ett 2D-plattformsspel med fokus på utmanande banor, smidig karaktärsrörelse och dynamisk, atmosfärisk värld. Spelet kommer att inkludera karaktärsutveckling, nivådesign och ett progressionssystem där spelaren kan låsa upp nya förmågor och möta ökande svårighetsgrad. Projektet struktureras i flera sprintar, där varje sprint levererar nyckelkomponenter.

Planeringsfas:

Innan utvecklingen börjar bryts projektet ner i hanterbara delar. Här är de övergripande kraven:

- Grundläggande spelmekanik:**
 - Karaktärens rörelser (gå, hoppa, dubbelhopp).
 - Plattformsmekanik (interaktion med plattformar och hinder).
 - Enkelt stridssystem med attack- och försvarsmekanik.
- Nivådesign och progression:**
 - Skapande av nivåer med ökande svårighetsgrad och komplexa pussel.
 - Introducera progression genom upplåsbara förmågor (t.ex. dubbelhopp, dash).
 - Pusselmekanik som kräver att spelaren löser gåtor för att komma vidare.
- Karaktärsutveckling och fiender:**
 - Skapa fiender med unika attackmönster och AI som anpassar sig till spelarens strategi.
 - Progressionssystem som belönar spelaren med nya förmågor efter framsteg.
- Användargränssnitt (UI) och UX:**
 - Skapa ett intuitivt gränssnitt för att navigera nivåer, samla poäng och visa Hp-bars.
 - Implementera poängräknare och nivåavslutningsskärmar.
- Berättelse och atmosfär:**
 - Introducera ett dialogsystem och interaktiva NPC för att förmedla berättelsen.
 - Skapa en visuell atmosfär genom ljudeffekter och bakgrundsmusik.
- Säkerhet och distribution:**
 - Säkra sparfiler och skydd av spelarens framsteg genom kryptering och autentisering.
 - Distribuera spelet via Steam eller Itch.io.

Dela upp arbetet i sprintar

Projektet delas upp i sprintar på 2 veckor. Varje sprint fokuserar på att leverera en färdig komponent som prioriteras baserat på spelvärde och utvecklingskomplexitet.

Iterationer (Sprintar):

Iteration 1: Grundläggande spelmekanik och rörelsesystem

Målet med den första iterationen är att få igång grunderna för spelet, där karaktärer kan röra sig och interagera i en enklare spelmiljö.

- **Utveckla grundläggande spelmekanik:**

Målet är att implementera grundläggande spelmekanik för att skapa en fungerande spelprototyp.

- Utveckla karaktärsrörelse: Skapa mekanik för att gå, hoppa och dubbelhoppa.
- Bygg en tidig nivåprototyp: En enkel karta med plattformar och hinder.
- Enhetstester: Säkerställ att rörelse och interaktioner fungerar som förväntat.

Iteration 2: Fiender och strid

Efter att ha etablerat grunderna i spelet är nästa steg att implementera stridssystemet och möjliggöra interaktioner med fiender.

- **Utveckla fiende-AI:** Skapa fiender med olika attackmönster och rörelser.
- **Implementera stridsmekanik:** Låt spelaren attackera och försvara sig.
- **Integrationstester:** Testa strider och fiendens AI för att säkerställa att interaktionerna fungerar.

Iteration 3: Utökad karaktärsdesign och balans

Nu när stridssystemet är på plats, utökar vi spelet med fler utmanande nivåer och pusselmekanik.

- **Bygg nya nivåer:** Lägg till hinder och pusselmekanik (dörrar, tryckplattor, etc.).
- **Implementera pussellösningar:** Integrera pussel som är beroende av spelarens rörelser och förmågor.
- **Användartester:** Utför användartester för att utvärdera nivådesign och pusslens svårighetsgrad.

Iteration 4: Berättelse och atmosfär

Fokusera på att integrera berättande element och skapa en mer engagerande atmosfär.

- **Implementera dialogsystem:** Introducera NPC som spelaren kan interagera med för att få ledtrådar och berättelseelement.
- **Lägg till atmosfäriska ljud och bakgrundsmusik:** som dynamiskt reagerar på spelarens rörelser och spelets tempo.
- **Användartester:** Testa hur berättelsen och atmosfären tas emot av spelarna.

Testning och distribution:

- **Enhetstester:** Efter varje iteration verifieras att de nya funktionerna fungerar och inte bryter mot spelets kärnfunktioner.
- **Integrationstester:** Vid slutet av varje sprint utförs integrationstester för att säkerställa att alla delar av spelet fungerar tillsammans.
- **Användartester:** En betatestgrupp får tillgång till spelet för att identifiera eventuella problem och balanseringsfrågor.
- **Inkrementell distribution:** Spelet distribueras successivt till plattformar som Steam och Itch.io för att samla in feedback innan slutlig lansering.

Underhåll och vidareutveckling:

Efter lansering kommer spelet att underhållas med regelbundna uppdateringar baserat på spelarfeedback, buggfixar och nya funktioner. Uppdateringarna kan inkludera utökad innehåll, nya nivåer, och förbättringar av spelmekanik för att hålla spelet engagerande för spelare.

- **Spelarfeedback:** Feedback samlas in genom in-game rapporter och online-forum för att prioritera framtida uppdateringar.
- **Bugfixar och prestandaförbättringar:** Regelbundna buggfixar och optimeringar baserat på rapporterade problem från spelare.
- **Innehållsuppdateringar:** Nya nivåer, fiender, och förmågor adderas för att hålla spelet fräscht och utmanande.
- **Löpande innehållsdistribution:** Expansioner och nedladdningsbart innehåll (DLC) distribueras via plattformar som Steam och Itch.io för att hålla spelet levande.

Nyckelkomponenter:

- **Spelmotor:** Unity eller Godot – Ideala motorer för 2D-plattformsspel, med kraftfulla verktyg för nivådesign och animering.
- **Grafik och ljud:** Aseprite för pixelart och FMOD för att skapa dynamiska ljudeffekter och musik.
- **Backend och data:** Firebase för att lagra spelarprogression, statistik och användarautentisering.
- **Säkerhet:** SSL/TLS för att säkra spelarens sparade data och framsteg.
- **Distribution:** Steam och Itch.io för att distribuera spelet och samla in spelarfeedback.