

1. **Programbeskrivning.** Beskriv detaljerat vad programmet gör.

Programmet utgör ett spel som går ut på att navigera en spelarstyrd karaktär igenom en bana till en utgångspunkt som finns utplacerad. I banorna så finns det olika block som kommer att påverka gravitation på olika sätt när spelaren kolliderar med dem. Färgade block kommer att rotera hela banan så att sidan med motsvarande färg är neråt medans svarta block får spelarens gravitation att ändras så att spelaren "faller" emot hållet han kolliderade med det svarta blocket. Det kommer även finnas andra mer negativt inriktade block i banorna som motarbetar spelaren igenom att t.ex. döda karaktären.

När man startar programmet blir man presenterad av en meny där man kan välja ifall om man ska köra en tutorial eller om man ska köra campaignen.

I tutorialen så kör man en bana där man stegvis introducerar nya element och förklarade text över hur dessa element fungerar för att ge spelaren en förståelig och interaktiv förklaring över hur spelet fungerar.

I campaignen har man val av att spela olika upplåsta banor; fler och fler banor låses upp medans man klarar tidigare banor. Under dessa tas tiden och så kan man samla in olika föremål för att få poäng för varje bana. Man kan gå tillbaka och spela om tidigare klarade banor för att försöka förbättra sin poäng.

2. **Användarbeskrivning.** Vem kommer att använda ert program? Vilka antaganden gör ni om användarna? Är de vana datoranvändare, är de specialister, nybörjare, små barn, etc.

Programmet kommer att vara designat för att användas utav relativt datorvana 15 till 25 åringar utan några fysiska nedsättningar. Svårighetsgraden på spelet kommer att vara mer anpassade till individer som har relativt stor datorspelsvana eftersom att spelet har ett väldigt pussellikt element i att man kan påverka banan igenom att rotera den samt att den förändrade gravitationen gör kontrol av karaktären relativt svår då under vissa förhållanden så kan en del kontroller agera inverterade.

3. **Användarscenarier.** Ge minst två exempel på scenarier där en av era tänkta användare använder programmet. Beskriv **i detalj** vad de ser, vilken typ av input de måste ge, hur de ger sin input och hur programmets output ser ut.

Scenario 1: Tutorial

Användaren startar programmet och blir presenterad av en meny. Hen klickar med vänster musknapp på tutorial knappen som finns i menyn. En bana startar och en informerar användaren om kontrollerna igenom text och tillhörande bilder som illustrerar vilken knapp hör till vad. Användaren måste sedan röra på karaktären för att gå vidare (vilket görs via, W/A/S/D). Sedan kommer text förklara hur ett element fungerar (färgade block etc.) och skapa en instans av sagd element. Spelaren måste sedan interagera med objekten igenom att kollidera för att gå vidare till nästa element. Efter användaren kört klart tutorialen återgår man till menyn där användaren väljer att stänga ner programmet.

Scenario 2: Campaign

Användaren startar programmet och blir presenterad av en meny. Hen klickar med vänster musknapp på campaign knappen som finns i menyn. Programmet presenterar en lista med numrerade knappar som representerar varsin bana, alla banor förutom bana 1 är markerade som låsta.. Användaren vänsterklickar på bana 1 och kör igenom den. Efter användaren klarat banan så presenteras hen av att val av att antingen gå vidare till

nästa bana eller återvända till menyn. Användaren är inte nöjd med sin poäng och går tillbaka till menyn och väljer bana 1 igen för att förbättra sin poäng. Efter banan denna gång så går användaren till menyn och stänger ner programmet.

4. **Testplan.** Beskriv hur ni tänker testa programmet. I den här uppgiften ska ni lägga extra vikt vid **användartestning**. Beskriv vilka uppgifter som testanvändaren ska utföra. De två användarscenarierna ska ingå i användartestningen.

Först och främst tänker vi skriva utförlig testkod så att vi kan hitta buggar och olika situationer där programmet kan krascha för att sedan fixa så att testkoden körs utan problem.

Sedan tänker vi använda en test grupp på så många personer vi kan få tillgång till (helst minst 10 personer). Där vi kommer att instruera dom att utföra först **Scenario 1** och sedan **Scenario 2**. Dock så kommer vi inte att förklara scenarierna i detalj utan för scenario 1 kommer vi be dom att "kör tutorialen" och för scenario 2 "kör bana 1 två gånger" för att se ifall att upplägget på programmet är intuitiv och förståelig.

5. **Programdesign.** Beskriv de grundläggande klasserna som ni avser att implementera och ge en beskrivning av de viktigaste metoderna i varje klass.

1. Game klass som agerar som behållare för hela programmet.
2. GameState klass som hanterar spel-staten. Dvs vad som händer när man faktiskt spelar spelet.
3. TutorialState ärver GameState för att få tag i all spellogik men innehåller egen kod för att skapa tutorial-specifika saker.
4. MenuState klass som hanterar startmenyn. Potentiellt flera menustates beroende på behovet av submenyer.
5. Map klass som hanterar konstruktionen och utritandet av banorna.
6. MapReader klass som klarar av att läsa ifrån fil och skicka vidare den informationen på ett sätt som Map klarar av att använda.
7. GameObject klass som hanterar allt generellt vad gäller spelobjekt.
8. Animation klass som hanterar animation inom GameObject.
9. Player klass som ärver GameObject med kod anpassad för spelarkarakteren.
10. Block klass som ärver GameObject som har kod för att konstruera diverse olika block som behövs.
11. Event klass för hantering av event ifrån input och output.
12. Partikelsystems klass som hanterar partiklar och hur dom ritas ut.

6. **Tekniska frågor.** En lista av tekniska frågor som ni måste hantera när ni bygger ert system. Var så detaljerad som möjligt. Ett viktigt steg mot en god design är att få ner så många frågor som möjligt på papper på ett organiserat sätt med så många förslag till lösningar som möjligt.

Hur fixar vi ett fungerande tutorial? Räcker det att bara ärva GameState och lägga in lite ny kod?

Finns det något enkelt och bra sätt att implementera partikelsystem modulärt?

Hur ska vi fixa rotationen på ett bra sätt?

7. **Arbetsplan.** Beskriv hur arbetet kommer att delas upp mellan personerna i projektet. Gör en tidsplan som visar när olika delmoment i projektet ska vara klara.

Datumet visar sista dagen då det är tänkt att arbeta på den givna uppgiften. Då person är klar innan utsatt tid är planen att antingen gå tillbaka och förbättra tidigare kod eller att försöka utveckla spelet igenom t.ex. att introducera nya block typer.

Datum	Erik Ihrén	Tobias Hamerslag
27/4	System för states	Events
28/4	GameState & MenuState	GameObject, Player & Block klasserna
29/4	Rotations algoritmen	Map & MapReader
30/4	Ev. modifieringar av tidigare kod för att fungera bättre med de nya tilläggen	Ev. modifieringar av tidigare kod för att fungera bättre med de nya tilläggen
2/5	Partiklar	Animationer
3/5	INDA Lektion	INDA Lektion
5/5	Campaign banor	TutorialState + tutorial banan
7/5	Användartestning + testkod	Användartestning + testkod
9/5	Bugfixning / Modifieringar	Bugfixning / Modifieringar