**Hand in #2**

**Kirsten, Daniel, Anne, Sebastian M, og Oskar A**

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, dokument

Automatisk genereret beskrivelse

**Formålet med vores spil:**

Formålet med spillet er at gøre spilleren opmærksom på vigtigheden og rutinen i at stille bordene på plads i klasselokalerne på VIA. Derfor er vores setting i spillet et klasselokale, spillets sprites er henholdsvis den karakter man styre med piletasterne, vores player, og de borde man kan flytte rundt på, vores projektiles. Her er det vores players opgave at flytte bordene hen på den plads der er blevet vist i starten af det level.

**Skitse til spillet:**

A white paper with black and white drawings

Description automatically generated

**Liste af smart requirements:**

1. Der skal være en Sprite (Player 1)
2. Denne skal kunne rykkes piletasterne
3. Spriten skal kunne flytte bordene.
4. Skal være målbart: Timer/countdown på 1 minut pr. Bane.
5. Spillet skal bestå af fem baner (levels), hvor sværhedsgraden stiger.
6. “Smart” win/lose scenarie -> spillet har et facit af hvordan bordene skal stå. Hvis de ikke står sådan når timeren går, taber man og kan genspille banen fra level
7. Skal være lærerigt (Ikke-funktionelt)

**Brug og inddragelse af vandfaldsmodellen:**

Kravspecifikation: Vi startede med at finde ud af hvad vi gerne ville have at spillet skal kunne, og hvad vi gerne have med i spillet. Dette gjorde vi med de førnævnte SMART-krav, som vi havde opstillet i prioriteret rækkefølge. Vi undersøgte om de opfyldte kravene, og derefter gik vi i gang med at designe vores spil.

Design: Vi gik i gang med at finde ud af hvordan spillet skulle se ud, og hvordan man kan implementere vores krav på vores design. Dette indebar især at finde ud af hvordan klasseværelset skulle se ud, og hvordan man skal kunne skelne mellem forskellige tiles. Dette gøres for at spilleren har en bedre mulighed for at kunne memorere hvor vedkommende skal placere bordene. Her har vi tænkt at der både skulle være en afgrænsning så man kan se hvor ét tile slutter og en ny begynder, men også at der skal være noget mere overordnet der kan give vores tilemap noget dybde. Dette skal alt sammen være med til at spilleren kan differentiere mellem tiles og giver bedre forudsætninger for at placere bordene rigtigt.

Implementering og testing: Vi startede med at skabe vores sprite (Player 1). Herefter sørgede vi for at det er muligt at bevæge den med piletasterne. Når det var gjort, begyndte vi at placere et bord på vores tilemap, og sørgede for at det er muligt for spilleren at flytte bordene med Player 1. Da det var muligt at gøre dette, placerede vi flere borde, og sørgede for at man kunne rykke flere borde efter hinanden. Efter det var muligt, begyndte vi at skabe vores tilemap. Her sørgede vi for at skabe nogle vægge og et ”gulv” i klasseværelset på den måde som vi indtænke i designfasen.

**Koden bag ”Bordnanas”-spillet:**

****

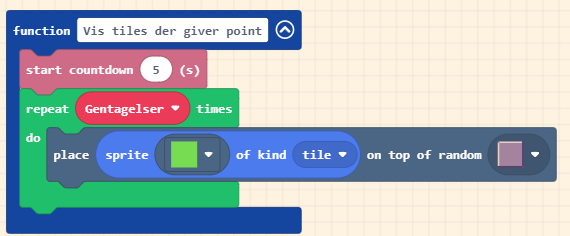
**On start:** Vi sætter vores character/sprite ved navn ”Eleven” til at være of kind player eftersom Eleven skal repræsentere spilleren. Eleven kan styres med piletasterne samtidig med at kameraet følger efter hans placering. Dernæst, indsættes det tilemap vi forinden har designet samt baggrund der passer til.

De første to variabler som sættes til ”False” bestemmer om spilleren ”Eleven” kan bevæge/move de borde der bliver lagt på banen i en anden. Denne kode kommer vi til senere.

”Move Projectile” og ”Move Bord” relatere til den samling af koder som gør det muligt for spriten ”Eleven” at samlet et af bordene op og placerer det igen. Disse koder kommer vi også til senere. Variablen ”Knapper virker” er blot for at sikre at spilleren ikke kan bruge knapperne inden spillet går i gang, som er når bordene bliver spawnet og de highlightede tiles er usynlige igen. Den sidste variabel ”Gentagelser” er blot den man skal ændre ud fra, hvor mange borde og dertil highlightede tiles, som skal benyttes på banen. Denne variabel er blot lavet for at man ikke skal ændre antallet flere forskellige steder, men kan nøjes med at gøre det et sted. I og med denne variabel er sat til 5, betyder det også at der spawner 5 borde og 5 highlightede tiles.

**Call function:**

Den sidste del af ”On start” kodeblokken, viser en call-function-blok og den function ser således ud:

****

Når spillet startes, sættes en countdown i gang. Den er sat til 5 sekunder og imens den tæller ned skabes der en grøn farvet tile of kind ”tile” x antal gange (tilsvarende værdien vi har givet ”gentagelser” variablen i On Start blokken”).

De er sat til at spawne oven på de eksisterende tiles vi bruger som gulv på vores bane. Ved at de spawner ”on top of random tile” så ser det pænt ud og risikere ikke den grønne tile ligger imellem to forskellige tiles.

**Countdown end:**



Eftersom vi startede en countdown i call-function blokken, så er der også sat en on countdown end blok lige under. Den gør først og fremmest at blokken i ”A” og ”B” knapperne nu virker, når man trykker på dem.

Derefter så spawnes der sprite of kind projectile (som skal forestille borde) x antal gange (tilsvarende værdien vi har givet ”gentagelser” variablen i On Start blokken”). Disse borde spawner oven på tilfældige eksisterende tiles, som vi bruger som gulv på vores bane.

Afslutningsvis så laver vi endnu en forlykke blok som referer til de grønne tiles of kind ”tile” som vi lavede i call-function blokken. De blokke får nu deres udseende ændret fra grøn til samme design som de andre tiles på vores bane har fra start.

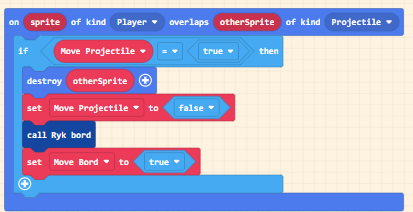
**Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse**

**”A” og ”B”- knapperne:** Her ser vi, hvordan knapperne bruges i spillet. ”A” benyttes til at samle bordene af type ”Projectile” op, hvilket opnås ved at sætte ”Move Projectile” til ”true”. (Som vi så tidligere i koden er den automatisk sat til ”false” når spillet åbnes.

”B”-knappen har til formål at slippe bordene af enten type ”Projectile” eller ” Bord”. Ligesom med A-knappen, virkker B-knappen heller ikke før ”knapper virker” variaben er true, hvilket den først bliver af sig selv efter countdown blokken senere i koden er færdig.

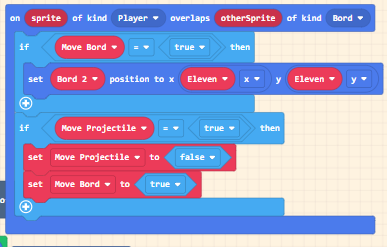
**OtherSprite of kind ”Projectile” event:**



Her er der startet med en event blok. Den venter på at vores sprite of kind player overlapper vores bord of kind projectile. Når dette sker, sætter der i gang i en logik blok. Her benyttes variablen ”move” igen. Når den er true (som den bliver når vi trykker på knappen ”A”) bliver othersprite of kind ”Projectile” ødelagt. Lige efter sættes ”Move” til false, så alle andre bruger vi overlapper efter ikke automatisk også bliver påvirket af dette event. Dernæst bliver der funktionen ”Ryk Bord” kaldt, som spawner en ny sprite of kind ”Bord”, med samme udseende som spriten of kind ”Projectile”. Funktionen gør blot det at den får den nye sprite til at spaw ved ”Elevens” koordinator.



**OtherSprite of kind ”Bord” event:**



Her er der startet med en event blok. Den venter på at vores ”sprite” of kind ”player” overlapper vores ”OtherSprite” of kind ”Bord”. Når dette sker, sætter der i gang i en logik blok. Her benyttes variablen ”Move Bord”. Når den er true (som den bliver i den sidste del af ”OtherSprite of kind Projectile event blokken”) bliver variablen ”Bord 2” sat til at følge ”Elevens” koordinator hele tiden, indtil ”Move Bord” er ”false” igen, hvilket kræver at spilleren trykker på ”B” knappen.

Hvis ”Move Projectile” er ”true” (som den kun er lige når man trykker på ”A” knappen) samtidig med ”Eleven” overlapper ”OtherSprite” of kind ”Bord”, så bliver ”Move Projectile” sat til ”false” og lige efter sættes ”Move Bord” til ”true”. Dette er blot for at fører dette event over til det forrige ”OtherSprite” of kind ”Projectile” event blok og gør at spilleren kan blive ved med at samle det sidste bord op, som ”Eleven” samlede op.

A screenshot of a video game

Description automatically generated

Koden herover styrer vores Sprite animation. Alt efter hvilken retning spriten bevæger sig i, vil animation gøre spillet mere realistisk.

A screenshot of a computer

Description automatically generated

I forlængelse af ovenstående, stopper alle animationer når Sprite står stille.

**OtherSprite of kind ”tile” event:**

**Et billede, der indeholder tekst, skærmbillede, nummer/tal, Font/skrifttype

Automatisk genereret beskrivelse**

For at gøre det muligt at vinde spillet gjorde vi sådan at når et af bordene overlapper en af de tidligere nævnte highlitedes tiles, efter at de ikke længere er grønne, så får spilleren 1 point samt den ændrer den highlitede tile til en anden type, så spilleren ikke bliver ved med at få flere og flere point imens ét bord er placeret på den samme tile.

Herefter har vi lavet en logik blok, hvor at hvis scoren er ligmed variablien ”Gentagelser”, som vi tidligere i koden fastsatte til 5, så vinder spilleren.