Boulder Dash Game Overview

BDRock

BDDiamond

**BDDoor**

BDEmpty

BDSand

BDWall

**BDPlayer**

BDBug

**IDBMovingObject**

**IDBObject**

**IDBKillable**

**AbstractBDObject**

**MyList**

**AbstractFallingObject**

**AbstractKillingObject**

**AbstractMovingObject**

**MapReader**

BoulderDashGUI

**MapReader**

BDMap

**MyGrid**

**MyList**

**Igrid**

**IList**

**OBS**:

* FallingTest funker ikke skikkelig. Av en eller annen grunn sluttet dette å virke 26.03.2017. Fungerte 24.03.2017.
* Når antall diamanter på map er 0 åpnes en dør. Fremtidig implementasjon vil være å laste inn et nytt map når spilleren når denne.
* Det er ikke implementert en AI for spilleren enda, men laget en enkel utvidelse av BDBug som gjør bevelsesmønsteret litt mer interessant.

**Systembeskrivelse**

- Beskriv hvordan programmet er organisert med dine egne ord. Dette kan også inkludere diagrammer for å beskrive strukturen til koden. Tegn gjerne på papir først mens du jobber med å få oversikt.

- Programmet har en grunnstruktur som blir bygget opp ifra BDMap. Her blir tillatte retninger definert og matrisen som spilleren beveger seg i blir også opprettet i denne klassen.BDMap tildeler også objekter til characters som blir brukt for å tegner kart. Sammen med MapReader tegnes kartet som spilleren navigere rundt på.

- Hvert objekt som finnes på kartet er definert ut ifra en klasse, og disse klassene har egne maler som forteller noe om hvilken oppførsel en kan forvente fra den type klassen, også kalt abstrakte klasser.

- I GUI delen blir alt som har med utseende til scenen definert. I tillegg er funksjonalitet for å avslutte spillet når antall diamater == 0 implementert. Dette kan utvides senere til å laste inn et nytt map.

- Hvert objekt som er en del av mappet arver fra en foreldreklasse, noen med mer avansert funksjonalitet en andre. Av ekstra effekter er det lagt til spiller animasjon, og lyd. Monstre har fått oppdatert bevegelse basert på tilfeldig retning, men er ganske begrenset og lite lur. Av andre ting er det implementert en funksjonalitet som er ment for utvidelse av spillet senere, ved at en dør kommer frem etter at alle diamantene er samlet inn.

1. Hvilke interface finnes? Hva er sammenhengen mellom dem og hva brukes de til?

- IGrid (metoder som kan benyttes for å manipulere et grid)

- IList (metoder som kan brukes for å manipulerer en liste)

- IGenerator (Brukes for å generere data til testing)

- IBDKillable (Identifiserer metoder for et objekt som kan dø i spillet)

- IBDMovingObject (Identifiserer metoder for et objekt som kan bevege seg i spillet)

- IBDObject (Identifiserer metodene til et objekt i spillet)

2. Hva slags rolle spiller arv i designet til programmet?

- Arv benyttes for å redusere duplikat kode ved å la objekt av samme typer arve fra egenskaper fra en mal.

3. Det er flere abstrakte klasser i systemet. Hva slags funksjon har de? Hvorfor er de abstrakte?

- Ved å bruke abstrakte klasser definerer vi et grunnleggende sett med oppførsler som det kan tenkes at alle typer av et objekt kan ha.

F.eks i AbstractFallingObject er det visse egenskaper som alle objekt som faller skal ha. Ved å la objektet arve fra den abrstakte klassen kan vi spare tid ved å gjenbruke kode. Vi kan også utvide koden ved behov.

4. Hvor er hoveddelen av logikken til spillet er implementert? Få oversikt over metodene, hvor de er implementert, hvordan de kalles.

- Hoveddelen av logikken er implementert under BDMap, BDPlayer, BDRock, BoulderDashGUI, samt under de abstrakte hovedklassene. Disse styrer egenskapene til objekter som BDRock, BDPlayer, BDSand, BDEmpty osv.

5. Hva slags rolle spiller abstraksjon i dette programmet?

- Abstraksjon i dette tilfellet danner grunnlaget for oppførselen til alle objekter som eksisterer på kartet.

6. Hvordan kunne man lagt til en ny type feltobjekter?

- Opprette nye klasser og kalle på disse. Se f.eks bruk av BDDoor

7. Hvordan er det implementert at en diamant faller?

- Implementert ved at den arver egenskapene til BDFallingObject.

Steg 3.1 Implementer getPosition(IBDObject obj) i BDMap

1. Hvorfor trenger vi getPosition-metoden? Kunne vi like gjerne ha lagret posisjonen i hvert enkelt objekt?

- Uten getPosition() vil det være tungvindt å holde styr på hvor et individuelt objekt befinner seg, uten å kartlegge hvert enkelt felt i matrisen, for hver oppdatering. Dette ville skapt utrolig mye unødvendig kode og gjort det vanskelig å hente ut informasjon. med getPosition() henter en ut data om et etterspurt objekt når det er nødvendig.

2. Hvis vi bruker et hashmap til å forenkle jobben med å finne posisjoner, har vi laget en sammenheng mellom to av feltvariablene; grid-et og hashmap-et. Hva er sammenhengen? (Her har vi en datainvariant, en begrensing på datarepresentasjonen i objektet.)

- Vi har da laget en sammenheng mellom Objektet opprettet i Grid og tildelt dette objektet en posisjon i et hashmap.

3. Ville vi fått en tilsvarende sammenheng mellom grid-et i BDMap og BD-objektene våre om vi lagret posisjonen i hvert enkelt objekt? Kan det være problematisk?

- Hvis vi gjorde dette, ville vi ikke hatt en indeksering på objektene våre. Ved hver iterasjon måtte vi da gått igjennom matrisen og sjekket hvor hvert enkelt objekt befant seg til enhver tid. Ved store matriser kan dette skape "lag".

**Images and Sound**

**Audio**:

All audio used are from <https://www.freesound.org>.

**Images**:

BDPlayer:

<https://jessebull.wordpress.com/tag/pixel-art/>

BDRock

<http://s444.photobucket.com/user/grandmadeb_rmvx/media/Granny_s%20Little%20Edits/VineyardA4RTPEditGrandmaDeb_zps4530c11d.png.html>

BDSand

<http://s444.photobucket.com/user/grandmadeb_rmvx/media/Granny_s%20Little%20Edits/VineyardA4RTPEditGrandmaDeb_zps4530c11d.png.html>

BDDiamond

<http://photobucket.com/gifs/pink%20diamond%20pixels>

BDBug

<http://www.ign.com/boards/threads/im-going-to-go-punish-the-slug.452934768/>

BDDoor

<http://s301.photobucket.com/user/IdanteUsharu/media/Door.gif.html>

BDWall

<http://s444.photobucket.com/user/grandmadeb_rmvx/media/Granny_s%20Little%20Edits/VineyardA4RTPEditGrandmaDeb_zps4530c11d.png.html>