

PG4400, vår 2016, innlevering:

# Breakout-clone

**Frist for innlevering:** søndag 17. april 2016, klokka 23:59.

**Innleveringsmåte:** "it's learning".

**Vurdering:** Innleveringen gis egen karakter, og denne teller 40% av emnekarakteren. Dere får tilbakemelding på innleveringen via its learning.

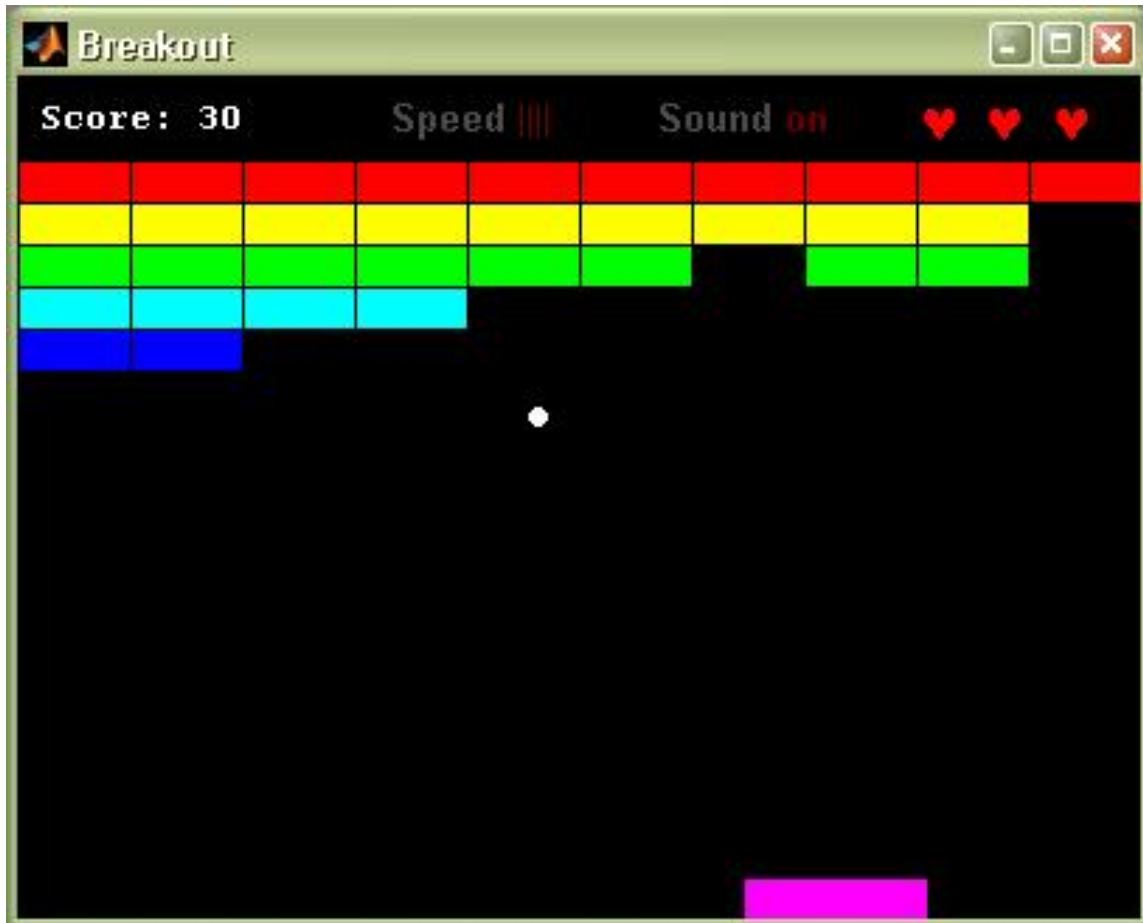
**Deltakere:** Innleveringen skal leveres i grupper på 2-3 personer, valgt av dere selv. Husk at innleveringen må vise hvem som er med på gruppa! Det anbefales å benytte "parprogrammering" (flere personer sitter sammen og skriver koden). Men ønsker man å jobbe alene, er det lov. NB - merk: Vurderingskriteriene er de samme, uavhengig av hvor mange som er på gruppa.

**Det legges vekt på:**

- Forståelse for C++. Ta med flest mulig av elementene vi har lært om. Inkluderer du noe som føles "kunstig", men som du har med for å vise at du behersker det, kommenter dette i følgedokumentet (se lenger ned.) Noen tips til C++ elementer: (trenger ikke benytte alle) exception handling, fil I/O, operator overloading, smart pointers, m.m. Husk også de små, lekre detaljene: const-by-reference parameteroverføring (der det er hensiktsmessig - ikke alltid!), const på alle fornufrige plasser, m.m.
- Ryddig kode. Gode navn på variabler og metoder, naming conventions og konsekvent språkbruk. (I den grad man benytter ferdig/utdelt kode trenger ikke denne følge samme standard som koden man skriver selv). Det er *ikke* nødvendig å bruke samme kodestandard som foreleser, så lenge deres egen er ryddig og gjennomført.
- God kode. Godt strukturert, objektorientert, logisk plassering av innhold.
- Kjørbart program og gode features. Se lenger ned for forslag til innhold i Breakout-clone.
- Følgedokument (pdf) der dere gjør rede for valgene og prosessen deres. Husk at dette er en del av sensuren: Det er viktig at dere forklarer løsningen deres for sensor! I dette tekstvedlegget, sørg for å få fram hvorfor akkurat deres løsning er så bra, dvs. vis (med ord, evt. figurer) at dere har tenkt, planlagt og ressonert dere frem til gode valg underveis for løsningen deres! Bruk også korte kommentarer i **koden** for å **bevisstgjøre** valgene deres.

**Merk:** Dersom dere får problemer med implementasjonen og ikke føler dere ferdige når dere leverer, påpek dette (problemene) i følgedokumentet nevnt over.

**Husk:** "Lekkert spill imponerer medstudentene, lekker kode imponerer læreren".  
Må du velge, prioritért riktig.



Kilde for bilde: [imgarcade.com/1/original-breakout-game/](http://imgarcade.com/1/original-breakout-game/)

## Muligheter og begrensninger

1. Det er lov å google etter tips – noe man også gjør i arbeidslivet.
2. Det som IKKE er lov – heller ikke i arbeidslivet – er å kopiere andres løsninger direkte! (Til info: Sensor kjenner flere Breakout-clones på nett.)
3. Det er imidlertid greit å benytte kode delt ut i timene/i forbindelse med øvingene. I så fall: Helt greit å bruke de 100% as-is, men bonuspoeng for å forbedre koden i de, og samtidig notere i følgedokumentet hva man har forbedret og hvorfor dette er en forbedring.

# Gameplay

**Basic:** Jeg tror alle kjenner til det grunnleggende Breakout konseptet?

- Man styrer en "paddle" frem og tilbake i bunnen av skjermen. Ideelt sett styres denne via keyboard input (styring via mus gjør det for enkelt å plassere paddle på rett sted).
- Ved start av spillet (og etter at man "dør") sitter ballen fast i paddle.
- Man trykker en knapp for å starte, og ballen fyres av gårde. Fra nå av sitter ikke ballen fast i paddle, men spretter tilbake opp i vinduet om den treffer paddle.
- Jo lenger ut på siden av paddle man treffer ballen, jo mer til siden spretter ballen.
- Om man bommer på ballen med paddle, slik at ballen faller ut i bunnen, mister man et liv/man dør.
- Målet med spillet er å treffe brikkene i toppen av brettet: Når en brikke treffes, fjernes den. Klarer man å fjerne alle brikkene, går man videre til neste brett.

## Tillegg

Elementene over er et minimum for å demonstrere spillfunksjonaliteten. Andre muligheter til innhold, som kan telle positivt på vurderingen om de gjøres riktig gjennom at man har mer C++ kode å briljere på, er bl.a.: (velg dere ut *noen*, ikke alle)

- **Menyer:** Main menu/oppstart bilde som ber brukeren trykke en tast når han er klar til å begynne. Tilsvarende et game over bilde som ber brukeren trykke en tast når han er klar for en ny runde.
- **Liv og Score:** Man får poeng hver gang en brikke fjernes. Man har flere liv som spillet holder styr på, og dør ikke før man har mistet siste liv.
- **Flere levels.** Spillet inneholder flere brett, og man avanserer til et nytt ved å fullføre det forrige.
- **Økende vanskelighetsgrad:** Hvert brett blir litt vanskeligere enn det forrige. Opp til dere å bestemme hvordan - kanskje ballen beveger seg litt fortere for hvert brett?
- **Power-ups.** Enkelte brikker som er tydelig merket (per egenskap) har spesielle power-up egenskaper om man treffer de. Opp til dere å velge hva power-ups gjør, men her er noen forslag: Ekstra bred paddle i x sekunder. En ekstra ball (og man mister ikke liv/dør ikke før begge er borte). "Laser" som kan skyte x skudd for å fjerne brikker, etc, etc, ...
- **Uknuselige brikker/vegger.** Enkelte brikker, tydelig merket, som ikke går i stykker om man treffer de. I stedet spretter ballen videre, som om de var vegger. (Disse telles selvfølgelig ikke med når man skal avgjøre om brettet er fullført eller ikke.)
- **Kule features** dere kommer på selv!

## Komme igang

For å få prosjektet opp og gå - lurt å gjøre følgende elementer TIDLIG i innleveringsperioden: (punktene er forklart i emnets forelesninger)

1. Via NuGet, importer SDL til prosjektet (og evt. SDL\_image, kanskje andre SDL extensions og).
2. Sjekk at Visual Studio og prosjektet fungerer fint med SDL.

## Grafikk:

Om dere ikke vil tegne selv: Google etter "breakout free game sprites", e.l.

## Tips til kode:

Dere står relativt fritt til å velge struktur, klasser, funksjoner, m.m. Her er imidlertid noen tips:

- Dere bør minimum lage egne klasser for:
  - GameManager
  - Paddle
  - Board (og/eller Level)
  - Piece ("brikke")
- Bruk en timer for å takle game loop frames av forskjellig lengde, som vi har snakket om i forelesninger:
- Lag en InputManager for å håndtere spillets input, som også snakket om i forelesningene.

## Til slutt:

Det er lov å gå gjennom problemstillinger dere måtte ha med foreleser og hverandre, inklusive via "spørsmål og svar" området på it's learning. Men del tips og linker til nyttige artikler, m.m. - *IKKE* del kode utenfor gruppene!

Lykke til, og kos dere! :-)