Innlevering 01 – Snake  
av Kristian Skarseth

# Oppgaven:

Lag et “snake” spill. Spilleren skal kontrollere en slange med piltastene. Målet er å plukke opp så mange “matbiter” som mulig uten å få “game over”¨

# Løsningen:

Jeg valgte å sette opp spillet så man først får opp en start meny når man kjører det. På startmenyen har man 3 valg:

* [P]Play
* [S]Settings
* [Q]Quit

Play kjører spillet, Settings åpner en settings meny og Quit avslutter spillet.

Om man trykker play uten å ha vært innom settings, vil spillet starte med default settings. Hvis man går innom Settings menyen og endrer noe der for så å gå tilbake til menyen og trykke Play, vil spillet kjøres med de endrede Settings.

På Settings menyen kan man velge mellom 3 pre-defined hastigheter, eller man kan sette hastigheten selv.  
Man kan toggle av og på “Dynamic speed”, som betyr at snaken går raskere hver gang man plukker opp en “food”.  
Man kan også toggle av og på Walls og Mines. Uten walls ender man opp på andre siden av vinduet istedenfor å få ” game over “ når man treffer veggen. Med mines, vil en “mine” spawne hver gang man plukker opp en “food”. Treffer man en mine vil man få “game over”

Man kan også Resette alle settings to default fra settings menyen.

Under selve spillet er det en liten meny på bunnen som med 2 setninger forklarer hvordan man kontrollere slangen, og hvordan man kan avslutte om man vil det. Det er også 4 indikatorer som viser om “walls” og “mines” er av eller på, hva scoren din er og hva speeden er.

Om man får “game over” vil spillet stoppe og en “game over” screen tegnes som viser scoren du fikk og info om hvordan man kommer tilbake til menyen.

# Valg o.l:

Da det er et forholdsvis lite prosjekt med lite kode, har jeg holdt meg til å kun skrive i GameManager istedenfor å opprette flere klasser.

Jeg valgte å bruke deque for X og Y koordinatene istedenfor list eller vector, da en deque er mer effektiv til de operasjonene den trengs til i dette spillet.

Jeg bruker ‘o’er for slangen sin kropp istedenfor ‘0’(null)er, da det så mer naturlig ut for meg. Om det er ønskelig kan man enkelt endre det ved å endre \_cSnakeBody /\_cSnakehead i \_defaultSettings().

Jeg har prøvd og inkludere alt vi har lært i timen så langt i løsningen, slik det nevnes i oppgaveteksten.

Jeg bruker \_ (understrek) foran private variabler. Jeg skriver også c for char, i for int, e for enum og b for bool som første bokstav i alle variabelnavn så de er enklere å kjenne igjen i koden.

# Forbedringer:

\*For mer oversiktlig kode kunne jeg brukt noen flere “for” loops til å printe ut tomme spaces istedenfor å gjøre det manuelt med printAt.

\*Forskjellig hastighet for å bevege seg opp og ned siden tegn er høyere enn de er brede. Det fører til at snaken beveger seg raskere oppover enn bortover. Jeg lagde en løsning på dette, men det fikk snaken til å lagge når man endret retning, så jeg lot være å ta det med.

\*Jeg burde laget spillet “size independant” ved å ha printAt/getCharAt etc bruke width/height delt på en verdi istedenfor hardkodede X og Y koordinater, slik at jeg kunne endret størrelsen på vinduet enklere.

\*Jeg har mens jeg jobbet kommet med ideer til hvordan jeg kunne gjort oppsettet av koden, samt logikken bedre. Jeg har noe kode som repeteres unødvendig. Hadde jeg gjort hele oppgaven på nytt nå, ville det nok blitt noe ryddigere og muligens mer effektiv kode. Noen variabler er også overflødige slik spillet er akuratt nå, men jeg lagde det med tanke på at det skulle være enkelt å legge til flere ekstra funksjoner og in-game settings.

\*Jeg har ikke brukt mye tid på å få farger på tekst perfekt, så det er en del steder jeg ville endret fargene om jeg hadde tid, men det har ingen ting å si for gameplayet.

Feil:

\*Slangen er ett tegn lengere første gangen man starter spillet. La merke til dette sent, og rakk ikke å finne en løsning på det. Det har derimot liten betydning for spillet.