



Westerdals

Oslo School of Arts,
Communication and Technology

Studieinformasjon 2017-2018

Bachelor i IT – Spillprogrammering
Avdeling for teknologi

Innhold

Innhold	i
1 Innhold	3
1.1 Introduksjon	4
1.2 Mål med studiet	4
1.3 Sentrale emner	5
1.4 Jobbmuligheter	5
1.5 Utveksling	6
1.6 Videre utdanning	6
1.7 Forkunnskaper	6
1.8 Undervisningsmåte	6
2 Emneplaner 1. studieår 2017-2018	7
2.1 DB1100 - Databaser 1	7
2.2 PGR100 - Objektorientert programmering	8
2.3 TK1100 - Digital teknologi	9
2.4 PRO100 – Kreativt webprosjekt	11
2.5 PGR101 - Objektorientert programmering 2	12
2.6 TK2100 - Informasjonssikkerhet	13
2.7 PRO101 Webprosjekt	14
2.8 VAL215 - Animasjon	15
2.9 VAL216 - Unity Utvikling	16
2.9 VAL217 - Digital kultur	17
3 Emneplaner 2. studieår 2017-2018	19
3.1 PG3400 Programmering i C for Linux	19
3.2 PG4200 Algoritmer og datastrukturer	20
3.3 Software design	22
3.4 PRO200 Smidig prosjekt	23
3.5 PG4400 C++ Programmering	24
3.6 RF3100 Matematikk og fysikk	25
3.7 PG4500 Game AI	27
4 Emneplaner 4. studieår 2017-2018	28
4.1 PG5200 Tools programmering	28
4.2 RF5100 Lineær algebra	29
4.3 PG6200 Grafikkprogrammering	30
4.4 PJ6100 Undersøkellesmetoder	31

4.5	BAO300 Bachelorprosjekt	32
4.6	PG5500 Embedded systems.....	33
4.7	PRO300 Virtual Reality prosjekt.....	34
4.8	ENT200 Entreprenørskap	35

1 Innhold

Dataspillbransjen har blitt en stor, internasjonal industri. Blant de best utdannede ansatte er programmererne. Disse har likevel sjelden en skreddersydd utdanning for produksjon av spill. Samtidig bruker flere og flere tradisjonelle industrier spill og spillrelatert teknologi.

Profesjonelle spill lages av grupper med variert størrelse, men noen roller er alltid med. Spillprogrammereren utvikler ny funksjonalitet og systemer i et spill. Forbedrete grafiske effekter, bedre kunstig intelligens og mer realistisk fysikk er oppgaver som programmereren typisk har ansvaret for. Et annet viktig arbeidsområde for spillprogrammerere er verktøy som andre utviklere bruker for å lage innhold i spillet. Med fokus på sanntids grafikksimuleringer har spillprogrammereren god innsikt i teoriene og teknologiene som brukes.

På Spillprogrammererstudiet vil studentene lære å realisere spennende og realistiske spillopplevelser. De får grundig opplæring i C++, så vel som andre programmeringsspråk som benyttes i profesjonell spillutvikling. Studentene lærer å integrere eksisterende teknologier til nye spill.

Studiet er 3-årig, og gir tittelen bachelor i IT. Programstrukturen er som vist nedenfor. Se også emnelisten i avsnitt 1.9.

Sem6	Undersøkellesmetoder (7,5 sp)	Bachelorprosjekt (22,5 sp)		
Sem5	Valgemne (7,5 sp)	Tools Programming (7,5 sp)	Lineær algebra (7,5 sp)	Grafikkprogrammering (7,5 sp)
Sem4	Smidig prosjekt (15 sp)	C++ programmering (7,5 sp)	Matematikk og fysikk (7,5 sp)	Game AI (7,5 sp)
Sem3		Programmering i C for Linux (7,5 sp)	Algoritmer og datastrukturer (7,5 sp)	Software design (7,5 sp)
Sem2	Valgemne (7,5 sp)	Objektorientert programmering 2 (7,5 sp)	Informasjonssikkerhet (7,5 sp)	Webprosjekt (7,5 sp)
Sem1	Databaser 1 (7,5 sp)	Objektorientert programmering 1 (7,5 sp)	Digital teknologi (7,5 sp)	Kreativt webprosjekt (7,5 sp)
	7,5 studiepoeng	7,5 studiepoeng	7,5 studiepoeng	7,5 studiepoeng

Gule farge forteller at emnet har felles undervisning med et eller flere andre studieprogram. Grønn farge viser plassering av valgemner. I 2. semester skal studentene velge et valgemne fra en samling valgbare emner på tvers av avdelinger og program i tillegg til de valgemnene som tilbys fra Avdeling for teknologi. Valg av emner skjer forutgående høstsemester.

Det første året er felles, og gir en solid grunnkompetanse i programmering, prosjektarbeid, systemutvikling, datateknikk og databaser.

Det andre året gir en grundig opplæring i programmeringsspråket C++, som et mye brukt språk for utvikling av store, moderne spill. C++ gir også en fin komplementering til Java som studentene lærer i første studieår: de behersker nå to utbredte programmeringsspråk. Forløperen til C++ er fortsatt i vid bruk, og studentene lærer C som en introduksjon til C++. Ved siden av C/C++ lærer studentene matematikk og fysikk, med fokus på temaer som er relevante for spill- og grafikkprogrammering. Som spesialiseringsemne innen spill lærer de Game AI (kunstig intelligens med fokus på spillbruk). Programmering involverer mer enn bare programmeringspråk, og studentene vil få et kurs i programmvarearkitektur som gir innsikt i bredere problemstillinger samt et emne innen algoritmer og datastrukturer, som forbedrer studentenes evne til å skrive ytelseseffektiv kode.

I det tredje året vil studentene jobbe med 3D grafikk, så vel som det matematiske grunnlaget for dette. Verktøyene studentene bruker er C++ og 3D systemet OpenGL. Studentene har også valgfag dette året, der de kan velge emner som gir de spisskompetanse innen spillindustrien eller som gir de breddekunnskap innen programmering generelt. I det avsluttende semesteret gjennomfører studentene et hovedprosjekt der de løser en krevende oppgave knyttet til spill- og/eller grafikkprogrammering, i nærkontakt med en privat eller offentlig virksomhet.

Studiet har i tillegg noen fellesemner som står sentralt i bachelor IT utdanningen. Disse emnene er Teknologi og Organisasjonen og Undersøkellesmetoder.

1.1 Introduksjon

Spillprogrammering: Utdanne programmerere som kan utvikle spill og tunge matematikk-/grafikkprogrammer

1.2 Mål med studiet

Bachelorprogrammet i IT med fordypning i spillprogrammering har som formålet å utdanne kandidater med kompetanse i å utvikle ny funksjonalitet og systemer i et spill, herunder en god kompetanse innen sanntids grafikksimuleringer og bred innsikt i teorier og teknologier som benyttes ved programmering av spill. Programmet vekt på bransjenært praktisk arbeid i tverrfaglige team gir studentene viktige erfaringer i arbeid med komplekse problemstillinger og en bred basis for å lykkes i deres videre karriere med å utvikle spill, forbedre grafiske effekter, bedre kunstig intelligens og skape mer realistisk i det grafiske uttrykket, samt med utvikling av verktøy for innholdsutviklere.

Etter endt studie skal kandidatene være i stand til å fylle et vidt spekter av programmeringsroller i norsk og internasjonalt næringsliv. De skal være høyaktuelle for spillindustrien, men de skal også være aktuelle som programutviklere innen C++ og Java generelt, spesielt i miljøer der visuelle virkemidler og multimedia benyttes samt tidskritiske applikasjoner. Kandidatene skal kjennetegnes av et læringsutbytte definert ved følgende kunnskaper, ferdigheter og generelle kompetanser:

Kunnskaper – kandidaten

- har bred kunnskap om programmering generelt og spesifikt innenfor språkene Java, C# og C++, som alle er relevante for spillprogrammering, et godt innblikk i programarkitektur, grafikkprogrammering, algoritmer og datastrukturer, samt anvendelse av matematikk, fysikk og kunstig intelligens for spill og simulatorer.
- har kjennskap til forsknings- og utviklingsarbeid innenfor spill- og multimedieprogrammering, samt i stand til å oppdatere sin kunnskap innenfor dette fagområdet

Ferdigheter - kandidaten

- kan benytte sine programmeringskunnskaper innen et bredt spekter av områder, fra å løse generelle programmeringsutfordringer innen objektorientering og programarkitektur, til mer spesialiserte utfordringer innen kunstig intelligens, simulasjon og grafikkprogrammering.
- har bred erfaring både innen software-verktøy, som det allsidige og kjente programmeringsverktøyet Visual Studio, og innen mer teoretiske verktøy som matematikk, for eksempel bruken av quaternioner i forbindelse med 3D-grafikk.

Generell kompetanse - kandidaten

- kan ta på seg varierte arbeidsoppgaver og programmeringsroller i norsk og internasjonalt næringsliv, slik at de er høyaktuelle for spillindustrien, men også aktuelle som programvareutviklere innen C#, C/C++ og Java, spesielt innenfor områder der simulasjon, visuelle virkemidler og/eller kunstig intelligens benyttes
- er i stand til å estimere, planlegge og gjennomføre større programmeringsoppgaver, både alene og som del av agile team (scrum), herunder i stand til både å benytte versjonskontrollprogrammer og prosjektstyringsverktøy for denne prosessen

1.3 Sentrale emner

Bachelor studium IT med fordypning i Spillprogrammering har følgende fokus: Ved siden av vanlig programmerer-kunnskap som software engineering og algoritmer/datastrukturer

tilpasset bruksområdet, fordrer dette kjennskap til mer spesialiserte fagområder og forskningsfelt som lineær algebra og spesielle geometriske teknikker, numeriske tilnærminger til klassisk mekanikk, optimalisering av grafisk programvare og kunstig intelligens.

1.4 Jobbmuligheter

Generelt vil studiet gi kompetanse til å gå inn i følgende roller:

Spillprogrammerer: Som spillprogrammerer jobber man daglig med å utvikle verktøy til dataspillbransjen og/eller kjernefunksjonalitet for dataspillene. Dette er en rolle som ligger svært tett opp til studiet. Man jobber i et ungt, spennende og hektisk miljø.

Systemprogrammerer: Mange firmaer lager programmer som krever mange av de samme egenskapene som spillprogrammering. Simuleringer, kontrollsystemer og andre tidsfølsomme applikasjoner pruser mange av de samme teknikkene. Dette betyr at spillprogrammereren kan jobbe for et vidt spekter av bransjer.

1.5 Utveksling

Det er mulig å ta deler av studiet ved andre høyskoler og universiteter i tråd med ERASMUS utvekslings program.

1.6 Videre utdanning

Etter fullført bachelor studium er det mulig å fortsette på en masterutdanning. Følgende retninger vil passe for studenter som har tatt bachelor studium IT med fordypning i Spillprogrammering:

- Retninger innen spill- og underholdningsprogrammering
- Teknisk programmering
- Prosjekt- eller IT-ledelse

1.7 Forkunnskaper

Studiet krever generell studiekompetanse med matematikkfordypning R1 eller S1 + S2 (2MX eller 2MZ + 3MZ) for å ta Bachelor IT: Spillprogrammering.

1.8 Undervisningsmåte

Studiet benytter en rekke undervisningsformer for å legge til rette for læring hos studentene. Forelesninger, presentasjoner, arbeidsmøte og øvinger brukes for å introdusere nye begreper og sammenhenger, samt innøve ferdigheter og forståelse. Arbeid med fagstoff foregår både individuelt og i gruppe, med vekt på samhandling for å oppnå et godt læringsutbytte. Studiet har prosjektarbeid alle semestre for å i praksis anvende ferdigheter og forståelse til å løse mer komplekse problemstillinger.

I andre og tredje studieår er problemstillingene sterkt knyttet til det enkelte studieprogrammets egenart. Studiet avsluttes med et hovedprosjekt i bedrift, hvor studentene vil anvende alle sine kompetanser opparbeidet gjennom studiet for å løse et problem/oppgave som har verdi for oppdragsgiver.

2 Emneplaner 1. studieår 2017-2018

2.1 DB1100 - Databaser 1

Engelsk navn: Databases 1

Studiepoeng: 7,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktivt design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: Ingen.

Anbefalte forkunnskaper: Ingen

Undervisning: Høst

Emneansvarlig: Per Lauvås

Innhold

Etter å fullført Databaser 1 skal man kunne forklare hva en relasjonsdatabase er, hva den kan brukes til og hvordan den skiller seg fra andre former for persistent lagring. Man skal kunne modellere og strukturere data for et domene. Man skal kunne opprette tabeller, legge inn ulike typer data i disse, kople dem sammen og hente ut rapporter ved hjelp av SQL spørringer. Man skal kunne forklare og anvende prinsippene for god design (normalisering, nøkkelbruk).

Læringsutbytte

Kunnskaper – kandidaten skal

- definere og forklare forskjellen på data, struktur og presentasjon
- forklare komponentene som inngår i et RDBMS
- definere hva ACID-egenskapene er og hvorfor disse bør oppfylles
- beskrive hva relasjonsalgebra er, og forklare mulighetene den gir ved å benytte begrepene kartesisk produkt, union, snitt og mengde
- definere begrepene primær-, fremmed-, kandidat- og super-nøkkel, og anvende disse i relasjonsdatabaser
- definere begrepene funksjonell avhengighet og normalisering
- forklare hvilke administrative mekanismer som finnes i et DBMS og hvordan disse brukes til å opprette databaser
- forklare roller og rettigheter i et DBMS
- forklare forskjellen mellom en relasjonsdatabase og en dokumentdatabase

Ferdigheter – kandidaten skal

- beskrive og utføre trinnene i utvikling av en EAR-modell fra kravspesifikasjonen til ferdig database, tegne en EAR-modell med korrekt notasjon og kardinalitet
- bruke SQL for å utføre CRUD-operasjoner på databasen
- bruke SELECT-spørringer med opsjoner til å hente rapport fra en enkelt tabell
- kombinere og hente rapporter fra flere tabeller ved bruk av JOIN-klausuler

- benytte de vanligste normaliseringsformene (UNF, 1NF, 2NF, 3NF, BCNF)
- bruke VIEW
- utføre enkel brukeradministrasjon
- modellere, opprette, legge inn data og hente ut rapporter fra en database med en viss kompleksitet (f.eks. 10-15 tabeller og 5-10 relasjoner)

Generell kompetanse – kandidaten skal

- forklare og eksemplifisere fordelene med å benytte en database og vurdere om disse er til stede i konkrete tilfeller
- skissere livsløpet til en typisk database: modellering, opprettelse, bruk, revisjon, og eventuell migrasjon.
- forklare hvilke lover og regler som gjelder for lagring og bruk av personopplysninger i Norge.
- vurdere kvalitetskriterier for korte videoer som er produsert for opplæringsformål innen informasjonsteknologi

2.2 PGR100 - Objektorientert programmering

Engelsk navn: Object Oriented Programming 1

Studiepoeng: 7,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktivt design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: Ingen

Undervisning: Høst

Emneansvarlig: Per Lauvås

Innhold

Emnet skal gi en første innføring i objektorientert programmering. Flere sentrale mekanismer i objektorientert programmering vil bli gjennomgått, samt eksempler på samspill mellom objekter av noen få klasser. Emnet skal også introdusere testing som en naturlig del av prosessen ved utvikling av kode.

Læringsutbytte

Kunnskaper – kandidaten skal

- vite hva et programmeringsspråk er, og hvordan slike skiller seg syntaktisk og semantisk fra naturlige språk
- forstå betydningen av sentrale begreper innenfor objektorientert programmering som for eksempel klasse, objekt og metode

- forstå livsløpet til et objekt
- forstå hva kompilering av kildekode innebærer
- vite hva variabler og typer er og hvordan de kan benyttes i programmering
- ha kjennskap til sentrale begreper knyttet til design av kode som kopling, samhörighet og innkapsling
- ha kjennskap til hvordan programmerere har hjulpet- og stadig hjelper hverandre ved å dele kunnskap gjennom ulike nettsamfunn.
- ha kunnskap om programmeringshistorie og hvilke paradigmer som har vært, og er, gjeldende innen fagområdet.

Ferdigheter – kandidaten skal

- beherske et enkelt integrert utviklingsverktøy (IDE) til å skrive og redigere kildekode samt kompilere og kjøre enkle programmer
- bruke de grunnleggende elementene i et objektorientert programmeringsspråk til å lage enkle programmer med enkle algoritmer
- benytte betingede og iterative kontrollstrukturer i et program
- bruke en debugger for å finne feil i kode
- lese, forstå og endre kode som andre har skrevet
- teste at kode fungerer som tiltenkt

Generell kompetanse – kandidaten skal

- redegjøre for hva objektorientert programmering innebærer

2.3 TK1100 - Digital teknologi

Engelsk navn: Digital Technology

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktivt design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: Ingen

Undervisning: Høst

Emneansvarlig: Bjørn Olav Listog

Innhold

For å kunne benytte en datamaskin på en effektiv måte må man vite hvordan informasjon kodes digitalt, samt hvordan den lagres, prosesseres og overføres av og mellom maskinvare og programvare. Ved å arbeide med emnet skal studenten lære seg å analysere datasystemer i ulike abstraksjonslag fra bit-nivå, via digitale kretser og maskinvarekomponenter (CPU, minne, busser og ulikt I/O-utstyr), data vs instruksjoner, operativsystem, applikasjoner og nettverkskommunikasjon. De skal kunne forklare hvordan

man med binærtall kan representere ulike former for informasjon. De skal erverve seg begrepsapparatet som trengs for å vurdere ulike maskin- og programvare opp mot hverandre. De skal kunne benytte modeller for funksjonell lagdeling i systemer, samt prosedyrer og verktøy til å forklare virkemåte og derigjennom kunne utføre effektiv feilsøking av enkeltmaskiner og nettverkskommunikasjon.

Læringsutbytte

Kunnskaper – kandidaten skal kunne

- beskrive den historiske utviklingen som ledet frem til dagens datamaskiner og nettverk
- definere og benytte de vanligste begrepene som benyttes i beskrivelse og analyse av datasystemer og nettverk
- prinsippene og metodene for digital koding av ulike former informasjon og hvilke muligheter og begrensninger som er forbundet med disse
- beskrive arkitekturen og organiseringen av datasystemer og nettverk
- forklare rollen til et operativsystem og dekomponere funksjonaliteten i hovedelementene prosess/tråd-administrasjon (scheduling + samtidshåndtering), minne-administrasjon (virtuelt minne mm), fil-administrasjon, I/O-administrasjon
- benytte TCP/IP-modellen til å forklare og analysere datakommunikasjon gjennom Internett

Ferdigheter – kandidaten skal kunne

- identifisere ulike kodingsformater for tekst, tall, lyd og bilder; kjenne virkemåten til disse og benytte dette i feilsøking
- med utgangspunkt i kjennskap til virkemåten til hovedkort, CPU, minne, I/O-kontrollere, busser og persistente lagringsmedia; identifisere de viktigste metrikker og benchmarks for ulike typer utstyr og identifisere mulige feilkilder og hvordan disse vil ytre seg
- benytte skall-kommandoer, overvåkings- og konfigurasjons- verktøy i Windows og/eller OSX/Linux
- benytte TCP/IP-modellen og kjennskap til tilhørende portokoller, til å analysere den fullstendige gangen i tilkopling til et LAN og nedlastingen av f.eks. en webisde

Generell kompetanse – kandidaten skal kunne

- gi en anbefaling ved innkjøp av datamaskin ut fra brukers behov og ønsker med bakgrunn i en korrekt og systematisk forståelse av tekniske spesifikasjoner for utstyret
- gjennomføre systematisk feilsøking og reparasjon av typiske problemer i hardware og software
- gi en begrunnet anbefaling ved valg av ISP og nettprodukt/abonnementstype, samt lokalt nettverksutstyr ut fra overslag på behov for bitrate og tjenestekvalitet
- benytte standard verktøy for å finne feil på og rette opp nettverksforbindelser

2.4 PRO100 – Kreativt webprosjekt

Engelsk navn:	Creative Web Project
Studiepoeng:	7,5
Fagområde:	Teknologi / IT
Undervisningsspråk:	NO / EN
Program:	Obligatorisk ved Bachelor i IT (alle studieprogrammer), Bachelor i digital markedsføring
Absolutte krav til forkunnskaper:	Ingen
Anbefalte forkunnskaper:	Ingen
Undervisning:	Høst
Emneansvarlig:	Eivind Brevik

Innhold

Studenten skal gjennom et prosjekt kunne benytte HTML- og CSS-teknikker for å kunne lage en interaktiv og kreativ løsning med animasjoner (CSS3-animasjon). Etter å ha fullført emnet skal studenten gjennom teamarbeid kunne utføre en kreativ prosess.

Læringsutbytte

Kunnskaper – kandidaten skal

- kunne forklare hva HTML og CSS er og hva forholdet mellom disse er
- ha kjennskap til ressurser på nett som tilbyr løsninger på problemstillinger i sammenheng med HTML og CSS
- kunne forklare hva kreativitet er
- kjenne til hvordan man arbeider effektivt i grupper

Ferdigheter – kandidaten skal

- kunne benytte HTML og CSS for strukturere og stilsette innhold etter etablerte retningslinjer
- kunne benytte HTML og CSS for å lage interaktive løsninger med animasjoner
- kunne gjøre en vurdering av gyldigheten til ressurser på nett som tilbyr løsninger på problemstillinger vedrørende HTML og CSS
- kunne anvende strukturert kreativ metode for problemløsning
- kunne analysere gruppeprosessen i et team

Generell kompetanse – kandidaten skal

- kunne anvende kreative teknikker og metode i et lite utviklingsprosjekt
- kunne redegjøre og reflektere over sine valg av løsninger
- kunne kritisk analysere og reflektere over gruppeprosesser i et team

2.5 PGR101 - Objektorientert programmering 2

Engelsk navn: Object Oriented Programming 2

Studiepoeng: 7,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktiv design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: Ingen.

Anbefalte forkunnskaper: PGR100 Objektorientert programmering

Undervisning: Vår

Emneansvarlig: Per Lauvås

Innhold

Emnet gir en videreføring i objektorientert programmering. Studenten kan definere og anvende spesialiseringer av klasser gjennom arv/interface/polymorfi og mestrer programmering av enkle grafiske brukergrensesnitt (GUI) med tilhørende unntakshåndtering. Studenten blir også introdusert til analyse og design ifm utvikling av kode, deriblant noen velkjente design patterns.

Læringsutbytte

Kunnskaper – kandidaten skal

- vite hva arv og polymorfi er
- vite om likheter/forskjeller mellom abstrakte klasser og interfaces
- vite hva unntakshåndtering innebærer
- vite hva et design pattern er, og kjenne til noen eksempler på slike

Ferdigheter – kandidaten skal

- definere spesialiseringer av klasser gjennom arv
- beherske bruk av abstrakte klasser og interface ('kontrakter')
- anvende arv og interfaces gjennom kode med polymorfisk oppbygning
- programmere enkle grafiske grensesnitt

Generell kompetanse – kandidaten skal

- forklare begrepet «teknisk gjeld» sett i lys av design og implementasjon av kode
- kunne forklare behovet for analyse og design ifm utvikling av kode

2.6 TK2100 - Informasjonssikkerhet

Engelsk navn: Information Security

Studiepoeng: 7,50

Fagområde: Teknologi / IT Ledelse; Spill; Film / Media

Undervisningsspråk: NO / EN

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktiv design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: TK1100 Digital teknologi, DB1100 Databaser 1, PGR100 Objektorientert programmering og PRO100 Kreativt webprosjekt

Undervisning: Vår

Emneansvarlig: Bjørn Olav Listog

Innhold

Trusselbildet for en datamaskinbruker er i dag preget av angrep fra datakriminelle som er ute etter direkte økonomisk gevinst, eller å overta enkeltmaskiner for å benytte disse videre til kriminell virksomhet. Bevissthet om de ulike trusslene som finnes i Internett er forutsetningen for å treffe riktige tiltak. Etter å ha fullført emnet skal en student være i stand til å analysere trusselbildet og foreta egnede sikringstiltak på egen maskin, i eget hjemmenettverk og gi begrunnede råd i forhold til oppsett og teknologivalg for websteder. Man skal også ha oversikt over hvilke lover og forskrifter som gjelder for bruk av datamaskiner til lagring, prosessering og formidling av data, her under personvern og opphavsrett

Læringsutbytte

Kunnskaper – kandidaten skal

- forklare hva som menes med informasjonssikkerhet.
- beskrive de ulike nivåene i informasjonssikkerhet med tilhørende trusselbilde og (mulige) tiltak.
- beskrive og forklare hvilke sikringstiltak som er tilgjengelig i ulike teknologier
- vurdere sikkerhetstiltak i forhold til gitte situasjoner

Ferdigheter – kandidaten skal

- utføre en risikoanalyse i forhold til en bestemt teknologisk løsning og foreslå tilpassede tiltak
- planlegge og gjennomføre (enkle) sikkerhetstester av bestemte teknologiske løsninger ved hjelp av standardverktøy

Generell kompetanse – kandidaten skal

- beskrive og følge lover og etiske normer som gjelder mhp opphavsrett og informasjonssikkerhet i Norge
- reflektere over sikkerhetsaspekter ved egen nett-adferd.

2.7 PRO101 Webprosjekt

Engelsk navn: Web Project

Studiepoeng: 7,50

Fagområde: Teknologi / IT Ledelse; Spill; Film / Media

Undervisningsspråk: NO / EN

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktiv design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: Ingen.

Anbefalte forkunnskaper: DB1100 Databaser 1, PGR100 Objektorientert programmering

Undervisning: Vår

Emneansvarlig: Siri Fagernes

Innhold

Etter å ha fullført emnet skal studenten kunne kjenne til ulike metoder for utvikling av programvare, og kunne gjennomføre et prosjekt etter en gitt prosjektmetodikk. Studenten skal kunne utforme en nettløsning i henhold til grunnleggende brukbarhetskrav med enkel databasefunksjonalitet.

Læringsutbytte

Kunnskaper – kandidaten skal

- kjenne til agile prosjektutviklingsmetoder
- kjenne til grunnleggende webdesign, grafisk grensesnitt og struktur
- ha kunnskap om retningslinjer og standarder for testing av webløsninger

Ferdigheter – kandidaten skal

- kunne gjennomføre grunnleggende webdesign, med vekt på prinsipper for brukervennlighet og universell utforming
- kunne utvikle et konsept ved hjelp av enkel prototyping
- kunne bruke et versjonshåndteringssystem for administrasjon av prosjektdokumentasjon (inkludert kildekode)
- kunne evaluere kvaliteten på et nettsted etter gjeldende retningslinjer og standarder
- kunne planlegge og gjennomføre et mindre webutviklingsprosjekt i team
- kunne dokumentere en løsning på en teknisk utfordring på en slik måte at dokumentasjonen kan være til nytte for andre.

Generell kompetanse – kandidaten skal

- kunne dokumentere og vurdere prosjekts resultatet etter gitte kriterier
- kunne dokumentere gjennomført prosjektprosess

2.8 VAL215 - Animasjon

Engelsk navn: Animation

Studiepoeng: 7,50

Fagområde: Teknologi / IT Ledelse; Spill; Film / Media

Undervisningsspråk: NO

Program: Valgemne ved alle studieprogram ved WOACT

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: Ingen

Undervisning: Vår

Emneansvarlig: Hrafnhildur Jonasdottir

Innhold

I dagens samfunn er det stadig en utvikling i teknologi, applikasjoner og teknikker. Animasjon knytter tradisjonelle teknikker som historiefortelling, tegning og konstruksjon sammen med ny teknologi innenfor kanaler som reklame, spill, film, mobiltelefoni, kunst, internett osv. Animasjon hjelper oss å kommunisere og visualisere våre budskap gjennom disse kanalene og spiller en stor rolle i utdanning, underholdning, industri og kunst. Hensikten med emnet er å tilegne seg kunnskap og ferdigheter til å lage en kort animasjonsfilm med lyd ved bruk av produksjonssoftware, samt vise forståelse innenfor dramaturgi og filmkunnskap til å kommunisere et gitt budskap gjennom visuelle grep. Studenten skal kunne vise til kreativ- og kritisk tenkning knyttet til animasjon.

Læringsutbytte

Kunnskaper – kandidaten skal

- forklare virkemåten til, og produksjonsmåten for, ulike animasjons-teknikker og –uttrykk
- beskrive rollen til manus, storyboard, lyd og etterarbeid i produksjon av (egen) animasjonsfilm
- gjengi og benytte regler og teori innenfor filmkunnskap og dramaturgi til å analysere animasjonsfilm
- beskrive de spesifikke visuelle virkemidlene som benyttes i animasjonskultur

Ferdigheter – kandidaten skal

- demonstrere ferdigheter innenfor tegning, fotografering, lyd og filmredigering gjennom å kunne skape en enkel animasjonsfilm fra ide til ferdig produkt
- demonstrere ulike teknikker for å lage en animasjonsfilm
- utvikle et eget animasjonsuttrykk
- benytte ulike digitale verktøy og filformater til å produsere, publisere og distribuere animasjoner.

Generell kompetanse – kandidaten skal

- demonstrere kreativ og kritisk tenkning og kommunikasjonsevner gjennom animasjonsteknikker

2.9 VAL216 - Unity Utvikling

Engelsk navn: Unity Development

Studiepoeng: 7,50

Fagområde: Teknologi / IT; Spill; Film / Media

Undervisningsspråk: NO

Program: Valgemne ved alle studieprogram ved WOACT

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: PRG100 Objektorientert programmering

Undervisning: Vår

Emneansvarlig: Tomas Sandnes

Innhold

Emnet omhandler programmering og utvikling med spillutviklingsplattformen Unity. Studentene blir kjent med utviklingsmiljøets muligheter og effektiv bruk av dette. Studentene lærer å importere, bruke og scripte forskjellige spillobjekter ("assets") i en "scene" for dermed å opprette "levels" med grunnleggende spillfunksjonalitet.

Læringsutbytte

Kunnskaper – kandidaten skal

- kunne grunnleggende C# syntaks.
- vite hvordan C# programmering kan brukes i Unity.
- forstå hensikten med en "game loop" og hvordan denne er nødvendig for spill.
- kjenne til Unitys event functions og bruksområdene for disse.
- forstå hensikten med local og world koordinater.
- kjenne til hvordan rotasjon, translasjon og skalering benyttes til å manipulere objekter.
- kjenne til hvordan kollisjoner implementeres i Unity.
- kjenne til hvordan mus, tastatur og touchskjerm kan benyttes for input.

Ferdigheter – kandidaten skal

- kunne benytte programmeringspråket C#.
- beherske Unity utviklingsmiljøet, kunne opprette prosjekter og publisere de til forskjellige plattformer.
- kunne legge til spillmekanikk ved hjelp av programmering.
- kunne skape/importere og anvende 3D-modeller, animasjoner, teksturer, lyder, "prefabs" og andre assets i Unity.

Generell kompetanse – kandidaten skal

- kunne planlegge, utvikle og publisere enkle spill.

2.10 VAL217 - Digital kultur

Engelsk navn: Digital Culture

Studiepoeng: 7,5

Fagområde: IT / Teknologi; Markedsføring; Kommunikasjon.

Undervisningsspråk: NO

Program: Obligatorisk ved Digital markedsføring. Valgemne ved alle øvrige studieprogram ved WOACT

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: Ingen

Undervisning: Vår

Emneansvarlig: Faltin Karlsen

Innhold

Internett har stor innflytelse på hvordan vi kommuniserer og våre liv, både privat og i kommersielle sammenhenger. Innen mange bransjer har Internett bidratt til å bryte ned skillet mellom produsent og tjenesteyter på den ene siden, og brukere og konsumenter på den andre. I dag påvirker brukerne ofte direkte hvordan nye produkter, tekster og tjenester utformes.

Samtidig gir Internett helt nye muligheter for å overvåke brukernes adferd og å styre tilgang til informasjon, i tillegg til at skillet mellom privat og offentlig blir mye mer uklart og det blir enklere for flere å delta som aktive brukere.

Emnet Digital kultur behandler disse temaene fra hovedsakelige humanistiske og medievitenskapelige kulturteoretiske tilnærminger gjennom fokus på generell kultur- og medieforståelse knyttet opp mot digitale kulturfenomener. Klassiske og moderne kulturteorier og tradisjoner benyttes i analyser og drøftinger av produsenter, innhold/tekster, forbrukere og kommunikasjonskanaler. Studentene lærer hvordan digitale kulturfenomener ligner på og skiller seg fra andre kulturfenomener, samt hvordan kulturteori kan anvendes for å kontekstualisere, belyse, diskutere og problematisere disse fenomenene.

Læringsutbytte

Kunnskaper

- Studentene skal kjenne til sentrale kulturteoretiske perspektiver og tradisjoner, og vite hvordan disse kan belyse forskjellige aspekter og problemstillinger rundt digitale kulturfenomener. Sentrale perspektiver og tradisjoner inkluderer semiotikk, retorikk, hermeneutikk, ideologikritikk og postmodernisme. Sentrale fenomener inkluderer sosiale medier, produsage, transmediering, franchising, kollektiv intelligens og spillkultur.
- Studentene skal kjenne til akademisk tenkning og skriving som sjanger, med fokus på analyse, drøfting, kildekritikk og referanseteknikk.

Ferdigheter

- Studentene skal kunne kritisk reflektere over og bruke fagstoff i analyser og drøftinger samt skrive akademiske tekster og holde faglige presentasjoner.
- Studentene skal kunne utvikle en strategi for bruk av sosiale medier i forretningsammenheng

Generell kompetanse

- Studentene skal kjenne til sentrale klassiske og moderne kulturteorier og kritisk anvende disse i forståelse og analyser av kulturfenomener
- Studentene skal kunne problematisere ulike tilnærminger rundt bruk av digitale medier i offentlig, kommersiell, og privat sammenheng
- Studentene skal kunne kritisk anvende fagstoff i egne akademiske tekster og presentasjoner som følger god akademisk skikk med hensyn til problemstilling, struktur, innhold, argumentasjon og referanseteknikk

3 Emneplaner 2. studieår 2017-2018

3.1 PG3400 Programmering i C for Linux

Engelsk navn: Programming in C for Linux

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO / EN hvis nødvendig

Program: Obligatorisk ved Bachelor i it - Spillprogrammering, Bachelor i it - Intelligente Systemer

Absolutte krav til forkunnskaper: PG2100 Programmering 2,-TK1100 Digital teknologi

Anbefalte forkunnskaper: Ingen

Undervisning: Høst

Emneansvarlig: Kjetil Raaen

Innhold

Arbeidet med dette emnet skal gi studenten innføring i programmeringsspråket C og hvordan dette kan brukes til å interagere direkte med operativsystemet. Det skal også gi forståelse av operativsystemet Linux og kunnskap om hvordan dette virker.

Læringsutbytte

Kunnskaper – kandidaten skal

- kjenne til operativsystemene Unix og Linux:
 - o bakgrunn og historie
 - o overordnet arkitektur og oppbygning
- kjenne til utviklingen av og bruksområdet for programmeringsspråket «C»
- forstå hva «virtualisering» er
- kjenne til byggeprosessen i C inkludert:
 - o compile
 - o link
- kunne forklare konseptet «everything is a file» og kjenne til eksempler inkludert:
 - o devices
 - o procfs
- forstå hva et «filsystem» er og hva det brukes til
- forstå konseptene bak brukere og permissions i Linux
- kjenne til forskjellen på userspace og kernelspace og hvordan bruke systemkall og signaler
- kjenne til minneområdene «stack» og «heap» og forklare forskjellen
- kjenne til konseptet "buffer overflow" og hvorfor det er farlig.
- forstå forskjellen på row-major og column-major arrays

Ferdigheter – kandidaten skal

- kunne bruke grunnleggende konstruksjoner i programmeringsspråket «C» inkludert:
 - o main, basic types, printf, scanf, strings, #include, include guards, enum, typedef, struct, union, command line parameters, macros, varargs
 - o Statisk minnehåndtering

- kunne utvikle programmer under paradigmet «strukturet programmering»
- kunne bruke stikkordet "const" korrekt
- kunne anvende dynamisk minnehåndtering inkludert:
 - o malloc, free, pointers, memcpy, sprintf, arrays, sizeof
- lese og skrive enkeltbits og grupper av bits in en større datastruktur.
- kunne generere kjørbare filer fra C-kode
 - o makefile
 - o static libraries
- kunne implementere lesing og skriving til fil
 - o r/w modes
 - o seek
 - o buffer size
- kunne operere kommandolinjen i linux for å utføre viktige operasjoner inkludert:
 - o ls, cd, mkdir, rm, cp, chmod, chown, pwd, ps, kill, man, wget
 - o pipes, redirection
 - o starte programmer
 - o mounting av filsystemer
- kunne lage programmer som unngår buffer overflow.

Generell kompetanse – kandidaten skal

- kunne vurdere hvilke situasjoner det er fornuftig å bruke programmeringspråket «C»
- kunne anvende «C» for å løse systemprogrammeringsoppgaver
- kunne forholde seg til manuell minnehåndtering
- kunne forholde seg til linux som operativsystem

3.2 PG4200 Algoritmer og datastrukturer

Engelsk navn: Algorithms and Data Structures

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor IT - Intelligente Systemer, Bachelor IT – Frontend- og mobilutvikling Bachelor IT - Programmering, Bachelor IT - Spillprogrammering

Absolutte krav til forkunnskaper: Gjennomført og bestått PGR100 Objektorientert programmering 1, PGR101 Objektorientert programmering 2.

Anbefalte forkunnskaper: Ingen

Undervisning: Høst

Emneansvarlig: Andrea Arcuri

Innhold

Emnet skal gi innsikt i algoritmer og datastrukturer som er sentrale i arbeidet med implementasjon og design av effektive datasystemer. Det legges vekt på asymptotisk analyse av worst-case ressursbruk, samt sentrale algoritmer og datastrukturer knyttet til søk og sortering. Emnet tar også for seg enkelte graf-algoritmer.

Læringsutbytte

Kunnskaper – kandidaten skal

- kjenne til sentrale abstrakte datatyper som lister, stakker, køer, prioritetskøer, mengder (*sets, collections*), symboltabeller (*maps*), trær og grafer
- kjenne til egenskapene til sentrale datastrukturer som tabeller (*arrays*), lenkede lister, heaps, binære trær, søketrær, balanserte søketrær, hashtabeller og grafer implementert ved naboskapslistor og -matriser.
- kjenne til sentrale søkealgoritmer som lineært søk, binært søk, søk i binære søketrær, søk i hashtabeller
- kjenne til sentrale sorteringsalgoritmer som Insertion sort, Selection sort, Bubble sort, Quicksort og Merge sort
- kjenne til metoder for traversering av grafer, samt Dijkstras algoritme, A*-algoritmen og Prims algoritme
- kjenne til oppbygning, virkemåte og bruk av rekursive funksjoner, inkludert rekursiv traversering av trær og grafer, rekursiv søk og sortering, backtracking
- kjenne til oppbygning, virkemåte og bruk av rekursive funksjoner, inkludert rekursiv traversering av trær og grafer, rekursiv søk og sortering, backtracking
- kjenne til problemstillinger knyttet til måling av kjøretid for dataprogrammer.

Ferdigheter – kandidaten skal

- kunne bruke kunnskapene nevnt i avsnittet over (kunnskapsmål) til å bruke eksisterende biblioteker for algoritmer og datastrukturer på en fornuftig måte
- kunne implementere kjente datastrukturer som tabell-lister, lenkede lister, binære søketrær, heaps, hash-tabeller og grafer
- kunne implementere kjente algoritmer som Insertion sort, Selection sort, Quicksort og Merge Sort.
- kunne bedømme worst-case ressursbruk for konkrete dataprogrammer ved hjelp av O-notasjon og tilde-notasjon
- beherske grunnleggende generisk programmering, samt bruken av grensesnitt (interface) og implementasjoner av grensesnitt, for effektiv bruk av algoritmer i kode.
- kunne sammenligne empiriske målinger av ressursbruk med teoretiske estimater, med tanke på å vurdere om estimatene er korrekte, å identifisere interessante avvik samt å anslå størrelsen på ukjente faktorer i estimatene

Generell kompetanse – kandidaten skal

- beherske klassisk asymptotisk analyse av dataprogrammer med O- og tilde-notasjon.
- kunne bruke eksisterende bibliotek og egenutviklede algoritmer og datastrukturer på en klok måte
- beherske terminologi som gjør det mulig å diskutere ressursbruk i dataprogrammer på en tilstrekkelig presis måte
- besitte kunnskap om algoritmer og datastrukturer som kommer til nytte i videre informatikkstudier og arbeidslivet

3.3 Software design

Engelsk navn: Software Design

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - Programmering og Bachelor i it – Spillprogrammering, valgemne ved Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Intelligente systemer.

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: PGR100 OO programmering 1, PGR101 Objektorientert programmering 2

Undervisning: Høst

Emneansvarlig: Tomas Sandnes

Innhold

Kurset skal gjøre studentene i stand til å designe og videreutvikle større programvaresystemer i tråd med kjente teknikker for modellering, testing og implementasjon.

Læringsutbytte

Kunnskaper - kandidaten skal

- Vite bakgrunn og innhold for UML standarden.
- Vite hva Unit Testing er.
- Forklare prinsippene for test-driven development (TDD).
- Kjenne til parprogrammering og hvordan bruken av dette påvirker softwareprosjekter.
- Kjenne til ofte brukte design guidelines og design patterns.
- Vite hva refactoring er.
- Vite hva multi-threading er.
- Forstå hvordan locking/lås kan brukes for å kode trådsikkert
- Forklare prinsippene for event handling.

Ferdigheter - kandidaten skal

- Ha oversikt over og kunne benytte den grunnleggende syntaksen i programmeringsspråket C# Beherske UML-diagrammene: use case, klassediagram og sekvensdiagram.
- Bruke UML til å designe programarkitektur.
- Kunne delta i parprogrammering på en produktiv måte.
- Kunne anvende design guidelines, som layers og følgende GRASP prinsipper: creator, information expert, low coupling, controller og high cohesion
- Kunne implementere følgende patterns: factory, decorator, MVC & MVP, singleton, composite, flyweight og façade.
- Kunne gjennomføre refactoring på eksisterende kode.
- Kunne implementere applikasjoner som benytter flere tråder.
- Beherske et integrert utviklingsverktøy (IDE).
- Skrive og redigere kildekode med dette verktøyet.

Generell kompetanse - kandidaten skal

- Reflektere over multi-threading og dets bruksområder.
- Samarbeide med andre programmerere for å utvikle gode programmer og for å videreutvikle sin kompetanse.
- Vurdere kvalitet på eksisterende programmer og vurdere strukturelle forbedringer.

3.4 PRO200 Smidig prosjekt

Engelsk navn: Argile Project

Studiepoeng: 15,00

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer; Bachelor i it - Interaktivt design; Bachelor i it – Mobilprogrammering; Bachelor i it – Programmering; Bachelor i it – Spillprogrammering; Bachelor i 3D-grafikk; Bachelor i digital markedsføring, Bachelor i spilldesign

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: Ingen

Undervisning: Høst og vår

Emneansvarlig: Per Lauvås

Innhold

Hensikten med emnet er å gi studenten en dypere erfaring i å mestre helheten i et større prosjekt, med vekt på anvendelse av en smidig metode: Scrum. Studenten skal planlegge og gjennomføre et omfattende prosjektcase i en tverrfaglig gruppe, og vil få trening å bruke moderne teknikker og verktøy.

Læringsutbytte

Kunnskaper – kandidaten skal

- ha kunnskap om smidig metodikk i et historisk perspektiv
- kjenne til hvordan smidige tilnærminger benyttes i ulike bransjer i arbeidslivet
- kjenne til ulike smidige metoder og rammeverk som eksisterer
- kunne redegjøre for hva begrepet «Design Thinking» innebærer
- kunne beskrive hensikten med og strukturen i Scrum
- kunne forklare hva en brukerhistorie er, og hva den brukes til
- kunne redegjøre for ulike måter å teste en løsning i et prosjekt
- kjenne til utfordringer og mulige løsninger for versjonering av ressurser i et prosjekt med flere deltakere

Ferdigheter – kandidaten skal

- kunne planlegge et prosjekt og utvikle en prosjektplan i tråd med Scrum

- kunne tilnærme seg og sitt team en problemstilling med bruk av «Design thinking»-prinsipper.
- kunne anvende et prosjektstyringsverktøy til planlegging og oppfølging av prosjektet
- kunne anbefale og håndtere versjonsering og arbeidsflyt knyttet til delte ressurser i et tverrfaglig prosjekt der flere deltakere er aktive samtidig

Generell kompetanse – kandidaten skal

- kunne samarbeide i en prosjektgruppe om planlegging og gjennomføring av en oppgave av betydelig størrelse som gjennomføres gjennom en lengre tidsperiode
- kunne identifisere mulige forbedringer i arbeidsprosesser gjennom en iterativ prosjektmetode
- kunne forklare smidig-begrepet til noen som ikke har kjennskap til hva begrepet innebærer

3.5 PG4400 C++ Programmering

Engelsk navn: C++ Programming

Studiepoeng: 7,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - Intelligente systemer, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: PGR101 Objektorientert programmering 2

Anbefalte forkunnskaper: PG3400 (Programmering i C for Linux

Undervisning: Vår

Emneansvarlig: Tomas Sandnes

Innhold

Emnet skal gi studentene kunnskap om fundamentale og avanserte programmeringsbegreper i språket C++. Målet er å videreutvikle studentenes programmeringskunnskaper til det som er nødvendig for å utvikle effektive og komplekse systemer, inklusive spill og interaktive applikasjoner. De vil lære å beherske Microsofts Visual Studio som IDE for C++.

Læringsutbytte

Kunnskaper – kandidaten skal

- kjenne til referanser og pekere, samt deres relasjon til hverandre
- kjenne til bibliotek for 2D grafikkprogrammering
- kjenne til grunnleggende konsepter i 2D grafikkprogrammering
- kjenne til minneallokering i C++, inklusiv forskjellene på stack og heap
- kjenne til function pointers i C++, samt callback functions og functors
- kjenne til templates og deres bruksområder
- kjenne til ulike revisjoner av C++ og de viktigste forskjellene

Ferdigheter – kandidaten skal

- beherske et egnet IDE som C++ programmeringsmiljø (opprette projects, kompilere, linke, kjøre)
- beherske debugging i et egnet IDE
- kunne programmere med standard I/O (cin, cout)
- beherske skriving og lesing til og fra fil med streams
- kunne anvende pekere og referanser
- beherske korrekt anvendelse av minne med new og delete
- beherske korrekt anvendelse av const og mutable
- kunne programmere med containers, algoritmer og iteratorer fra STL
- kunne anvende og utvikle egne templates
- kunne anvende objektorientert programmering i C++ med bruk av klasser, objekter, arv og polymorfi
- kunne programmere med threads og synkroniseringsmekanismer i C++
- kunne anvende smart pointers
- kunne programmere med operator-overlasting
- kunne håndtere og lage egne exceptions
- Kunne inkludere og anvende tredjepartsbiblioteker
- beherske enkel 2D grafikkprogrammering
- benytte callback functions
- kunne anvende tid som en faktor under kjøring av et program
- kunne håndtere brukerinnteraktivt i applikasjoner
- kunne utnytte "move semantics" for bedre kode
- kunne benytte tråder, promises og futures til parallell programmering

Generell kompetanse – kandidaten skal

- kunne utvide sin forståelse for hvordan objektorientering og template programming kan benyttes til å skape velldesignede programmer
- forstå poenget med operator-overlasting og bruksområde
- forstå når man skal bruke tredjeparts biblioteker og når en skal utvikle selv
- forstå hvordan parallell programmering er nødvendig for å få optimal ytelse av moderne hardware

3.6 RF3100 Matematikk og fysikk

Engelsk navn: Mathematics and Physics For Game Programming

Studiepoeng: 7,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: RI / Studierett ved program med R1-krav

Anbefalte forkunnskaper: Ingen

Undervisning: Vår

Emneansvarlig: Lars Sydnes

Innhold

Kurset gir en grunnleggende innføring i geometri, fysikk og matematikk som kommer til nytte i arbeid med grafikk og fysikkmodellering i forbindelse med spillprogrammering.

Læringsutbytte

Kunnskaper – Etter å ha fullført emnet skal studenten kjenne til:

- grunnleggende vektorgeometri
- grunnleggende bevegelseslære og klassisk fysikk, beskrevet ved Newtons lover og impulsloven

Ferdigheter – Etter å ha fullført emnet skal studenten beherske:

- beregninger basert på vektoroperasjoner, skalarprodukt og kryssprodukt og to- og tredimensjonal vektorgeometri
- beskrive geometriske objekter som linjer, plan, sirkler, sfærer og trekanter ved hjelp av ulike representasjoner som ligninger og parameterfremstilling
- bruke ligningsfremstillingen og parameterfremstillingen til å finne skjæringspunkter mellom geometriske objekter
- vinkelberegninger uttrykt med radianer og grader, utført ved hjelp av trigonometriske funksjoner og vektorgeometri
- omregning mellom kartesiske koordinater, polarkoordinater, sylinderkoordinater og sfæriske polarkoordinater
- analytiske bevegelsesberegninger basert på konstant akselerasjon
- numeriske bevegelsesberegninger basert på Newtons bevegelseslover
- aktuelle numeriske metoder er Eulers metode, Heuns metode og Størmer-Verlet metoden
- kollisjonsberegninger basert på impulsloven, både for elastiske og uelastiske kollisjoner

Generell kompetanse – Etter å ha fullført emnet skal studenten beherske:

- den vektorgeometrien som skal til for at man kan løse geometriske problemer som oppstår i forbindelse med programmering av 2D-grafikk
- teknikker i matematikk og fysikk som gjør det mulig å implementere grunnleggende fysikk i dataspill

3.7 PG4500 Game AI

Engelsk navn: Game AI

Studiepoeng: 7,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: PGR101 Objektorientert programmering

Anbefalte forkunnskaper: Ingen

Undervisning: Vår

Emneansvarlig: Kjetil Raaen

Innhold

Etter fullført emne har studentene fått en innføring i kunstig intelligens for spill: Hvilke byggesteiner som utgjør AI i ulike typer spill, og hvilken teknikk som passer i en bestemt situasjon. Videre har studentene fått praktisk erfaring med å implementere kunstig intelligens.

Læringsutbytte

Kunnskaper – kandidaten skal

- kjenne til forskjeller og likheter mellom kunstig intelligens (AI) for spill og mer tradisjonelle bruksområder.
- vite hva pathfinding er, samt kunne sammenlikne en rekke kjente pathfinding algoritmer.
- kjenne til hierarkisk pathfinding.
- kjenne til fuzzy logic, og hvordan dette kan brukes i AI sammenheng.
- vite hva regelbaserte systemer er, og hvordan de virker.
- kjenne til måter AI kan ta taktiske og strategiske avgjørelser på.
- vite hvordan vi kan lage koordinert AI for grupper.
- kjenne til teknikker for å la AI systemer lære, blant annet ved hjelp av nevrale nettverk.
- Kjenne til min-max tresøk med optimeringer

Ferdigheter – kandidaten skal

- kunne lage og bruke agentbasert AI.
- kjenne til steering behaviors, og bruke de vanligste steering behaviors i praksis.
- kunne lage og bruke tilstandsmaskiner, særlig i forbindelse med spill.
- kunne implementere riktig AI for spill innen flere sjangere.
- kunne programmere A* algoritmen med fler for pathfinding.
- kunne scripte AI-oppførsel for spill.

Generell kompetanse – kandidaten skal

- forstå hvordan AI kan brukes til å gi unike spillopplevelser.
- forstå hvilke AI-teknikker som er riktig å velge, gitt bestemte situasjoner.

4 Emneplaner 4. studieår 2017-2018

4.1 PG5200 Tools programmering

Engelsk navn: Tools Programming

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO / EN hvis nødvendig

Program: Obligatorisk ved Bachelor i it – Spillprogrammering, valgemne ved Programmering

Absolutte krav til forkunnskaper: PG2100 Programmering 2

Anbefalte forkunnskaper: Ingen

Undervisning: Høst

Emneansvarlig: Kjetil Raaen

Innhold

Arbeidet med dette emnet gir studentene innsikt i utfordringer å løsninger for utvikling av verktøy som lar andre profesjoner produsere innhold for spill. Studentene utvikler en enkel leveleditor og andre verktøy som trengs for å utvikle spill og andre multimedieprodukter. Videre vil studentene bli kjent med behovet for stabilitet, feilhåndtering og hurtig reaksjon.

Læringsutbytte

Kunnskaper – kandidaten skal

- forstå grunnleggende funksjonalitet i en spillmotor, og hva konseptet "toolchain" innebærer.
- kjenne til forskjellige typer spillmotorer samt forskjellen mellom rene grafikkmotorer og komplette spillmotorer.
- forstå hvorfor gode verktøy er essensielle for effektiv produksjon av innhold.
- kunne forklare de forskjellige kategoriene av verktøy brukt i spillutvikling inkludert leveleditors, propertyeditor og support tools.
- kunne beskrive den viktigste funksjonaliteten i en leveleditor.
- kjenne til tredjepartsløsninger for spillverktøy.
- kjenne til nettverksprotokoller og hvordan spill kommuniserer over nett.

Ferdigheter – kandidaten skal

- kunne vurdere nytten av å lage utviklingsverktøy selv i forhold til å benytte tredjepartsløsninger.
- kunne utvikle kravspesifikasjon for et variert utvalg verktøy.
- kunne programmere verktøy i C# med WPF.
- kunne implementere essensiell funksjonalitet for en leveleditor.
- beherske bruk og håndtering av exceptions for bedret stabilitet.
- kunne unngå tunge operasjoner i GUI-tråd.
- kunne legge inn support for scripting i eksisterende kode.
- kunne lagre spilltilstand og logge hendelser i en database.
- kunne serialisere og deserialisere spilldata.

Generell kompetanse – kandidaten skal

- forstå hvordan en god verktøykjede kan forbedre utviklingseffektiviteten.
- kunne reflektere kritisk rundt spillutviklingsprosessen og de nødvendige krav.

4.2 RF5100 Lineær algebra

Engelsk navn: Linear Algebra

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor IT - Spillprogrammering

Absolutte krav til forkunnskaper: RF3100 Matematikk og fysikk.

Anbefalte forkunnskaper: RF3100 Matematikk og fysikk

Undervisning: Høst **Emneansvarlig:** Lars Sydnes

Innhold

Dette kurset skal forberede studenten til å arbeide med tredimensjonal grafikkprogrammering, ved å gå gi nødvendig innsikt i geometri, transformasjonsteori og lineær algebra. Ved siden av dette er det et viktig mål at studenten skal få en generell trening i bruk av matematiske metoder.

Læringsutbytte

Kunnskaper – kandidaten skal

- kjenne til grunnleggende matriseregning inkludert addisjon, multiplikasjon, transponering, invertering og determinanter
- kjenne bruken av lineære og affine transformasjoner, med vekt på geometriske transformasjoner som translasjon og rotasjon
- kjenne til ulike beskrivelser av rotasjon og romlig orientering, som rotasjonsmatriser, Euler-vinkler, rotasjonsvektorer, akse-vinkel-representasjon og kvaternioner
- kjenne til metoder for å regne om mellom de nevnte beskrivelsene av rotasjon og romlig orientering
- kjenne til homogene vektorer og matriser, og spesielt matriser for sentralprojeksjon
- kjenne til metoder for løsning av lineære ligningssystemer, som Gauss-eliminering og matriseinversjon
- kjenne til bruk av skalarprodukt og matrise i forbindelse med projeksjoner og approksimasjoner og minste kvadraters metode
- kjenne til barysentriske koordinater og interpolasjon i trekanten, samt bruken av dette i 3D-grafikk, som for eksempel i forbindelse med Phong-shading og teksturering.
- være orientert om flere anvendelsesområder for lineær algebra, som for eksempel beregning av interpolerende kurver og lineær regresjon med minste kvadraters metode

Ferdigheter – kandidaten skal

- kunne løse geometriske problemer ved å bruke teorien for vektorer, matriser og lineære ligningssystemer, innenfor de feltene som er beskrevet i avnittet om kunnskapsmål
- kunne kombinere og veksle mellom ulike koordinatsystemer som modellkoordinater, verdenskoordinater, kamerakkoordinater og skjermkoordinater
- bruke lineær algebra på enkelte anvendelsesområder

Generell kompetanse – kandidaten skal

- ha god oversikt over vektorgeometri og grunnleggende lineær algebra med anvendelser, innenfor de områdene som er beskrevet i avsnittet om kunnskapsmål
- ha et godt utgangspunkt for å arbeide videre med grafikkprogrammering
- utvikle sin matematiske modenhet.

4.3 PG6200 Grafikkprogrammering

Engelsk navn: Graphics Programming

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO / EN (emnet vil bli undervist på engelsk etter behov)

Program: Obligatorisk ved Bachelor i it - spillprogrammering

Absolutte krav til forkunnskaper: PG4400 C++ Programmering, RF3100 Matematikk og fysikk

Anbefalte forkunnskaper: RF3100 Matematikk og fysikk

Undervisning: Høst

Emneansvarlig: Martin Lilleeng Sætra

Innhold

Arbeidet med dette emnet skal gi studenten en god oversikt over hvordan grafikk programmeres med C++ og moderne OpenGL, samt de sentrale teknikker og metoder som brukes. Emnet skal videre gi innsikt i avanserte metoder for grafikk og rendering (multipass-algoritmer, avansert lyssetting og skygger, mm.), blant annet ved bruk av nettopp OpenGL og OpenGL Shading Language. Videre lærer man om effektiv representasjon og behandling av geometri.

Læringsutbytte

Kunnskaper – kandidaten skal ha kunnskaper om blant annet

- representasjon av geometri,
- farger, lyseegenskaper og materialeegenskaper,
- buffer objects,
- framebuffer objects,
- multipass-algoritmer,
- per-pixel lyssetting,
- blending,
- teksturering,
- skygger,
- cube maps,
- normal-mapping
- subdivisjon

Ferdigheter – kandidaten skal kunne

- arbeide med geometriske transformasjoner,
- benytte kvaternioner, vektorer og matriser ifm. grafikkprogrammering og
- arbeide med ulike projeksjoner ifm. grafikkprogrammering.

Generell kompetanse – kandidaten skal

- kunne vise en god oversikt over OpenGL og OpenGL "state machine" (tilstandsmaskin).
- vise bred kunnskap om 3D-grafikkprogrammering, bruk av shadere og avanserte emner innen grafikk og rendering.
- vise kjennskap til hovedtrekk i utviklingen av OpenGL, samt utviklingen av grafikkmaskinvare og moderne grafikkort-arkitektur.
- være komfortabel med å programmere OpenGL i Visual Studio med C++.
- kunne vise til praktisk erfaring med hvordan OpenGL fungerer gjennom modifisering og utvidelse av utdelt «skjelettkode».

4.4 PJ6100 Undersøkellesmetoder

Engelsk navn: Research Methods

Studiepoeng: 7,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO /

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktiv design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering, Bachelor i digital markedsføring

Absolutte krav til forkunnskaper: Ingen.

Anbefalte forkunnskaper: Ingen

Undervisning: Vår

Emneansvarlig: Wanda Presthus

Innhold

Emnet skal gjøre studenten kjent med metoder med særlig vekt på metoder som har relevans for IT-bransjen. Emnet er et støtteemne til hovedprosjektet, primært i forhold til rapportskriving og metode.

Læringsutbytte

Kunnskaper – kandidaten skal kunne

- gjøre rede for forskjellen mellom kvalitative og kvantitative metoder
- gjøre rede for grunnprinsippene i metoder som intervjuer, spørreundersøkelser og observasjon

Ferdigheter – kandidaten skal kunne

- gjennomføre en undersøkelse som er relevant for et IT-prosjekt

- begrunne valg av metode ut fra en gitt faglig problemstilling
- skrive en metoderapport som kan inngå som del av BAO300-Bachelorprosjekt
- gjennomføre litteratursøk og utarbeide en korrekt referanseliste
- kunne formidle sine funn skriftlig

Generell kompetanse – kandidaten skal kunne

- vurdere og benytte relevant metodikk for et IT-prosjekt
- vurdere etiske forhold knyttet til undersøkelsesmetoder

4.5 BAO300 Bachelorprosjekt

Engelsk navn: Bachelor Thesis

Studiepoeng: 22,50

Fagområde: Teknologi / IT

Undervisningsspråk: NO / EN

Program: Obligatorisk ved Bachelor i it - E-business; Bachelor i it - Intelligente systemer, Bachelor i it - Interaktiv design, Bachelor i it – Frontend- og mobilutvikling, Bachelor i it – Programmering, Bachelor i it - Spillprogrammering

Absolutte krav til forkunnskaper: DB1100 Databaser 1, PGR100 Objektorientert programmering, TK1100 Digital teknologi, PRO100 Kreativt webprosjekt, PGR101 Objektorientert programmering 2, TK2100 Informasjonssikkerhet, PRO101 Webprosjekt, PJ3100 Project Software Engineering

Anbefalte forkunnskaper: Ingen

Undervisning: Vår

Emneansvarlig: Hanne Sørum

Innhold

Studenten skal få yrkeserfaring ved å gjennomføre et prosjekt i en bedrift, etablere eget selskap eller delta i forskningsprosjekt. Studenten skal demonstrere bred kunnskap om sentrale temaer og teorier, og vise ferdigheter i metoder, verktøy og teknologi innenfor fagområdet.

Læringsutbytte

Kunnskaper – kandidaten skal kunne

- forklare hvordan oppdraget er til nytte for virksomheten/bedriften/oppdragsgiver
- planlegge og styre gjennomføringen av prosjektet
- anvende kilder på en korrekt måte

Ferdigheter – kandidaten skal kunne

- innhente nødvendig informasjon på en strukturert måte
- planlegge og gjennomføre et prosjekt i en bedrift etter valgt metodikk
- utforme og kvalitetssikre løsningen i henhold til spesifikasjoner og etablerte standarder

- formidle sentralt fagstoff, teorier og løsninger både skriftlig, muntlig og gjennom andre relevante uttrykksformer

Generell kompetanse – kandidaten skal kunne

- beskrive og vurdere erfaringer i prosjektet, og hva som er lært av prosessen.
- formidle resultatet av prosjektet på en klar og velstrukturert måte, tilpasset målgruppen.
- vurdere hvordan oppdraget forholder seg til etablert kunnskap/forskning på området

4.6 PG5500 Embedded systems

Engelsk navn: Embedded Systems

Studiepoeng: 7,5

Fagområde: Teknologi / IT

Undervisningsspråk: NO

Program: Obligatorisk ved Bachelor i IT - Intelligente systemer. Valgemne ved Bachelor i IT – Spillprogrammering

Absolutte krav til forkunnskaper: Ingen

Anbefalte forkunnskaper: Ingen

Undervisning: Høst, Vår

Emneansvarlig: Tomas Sandnes

Innhold

Arbeidet med dette emnet skal gi studenten teoretisk og praktisk kunnskap om embedded systems med tilhørende arkitektur. Emnet skal også gi kunnskap om grunnleggende hardwarekomponenter og Arduino. Emnet skal gi studentene ferdigheter i å programmere applikasjoner som kommuniserer med komponenter tilknyttet mikrokontrollere og Unix-baserte maskiner. Studenten skal ha kompetanse i bruksområder og praktiske anvendelser for embedded systems.

Læringsutbytte

Kunnskaper – kandidaten skal

- Forstå uttrykket "embedded systems" og kjenne til forskjellen mellom en CPU og en microcontroller.
- Kjenne til de mest brukte arkiturene for embedded systems.
- Ha kunnskap om mikrokontrollere som Arduino og hvordan disse brukes.
- Ha kunnskap om hardwaren i Raspberry Pi, hvordan den brukes og hvordan den skiller seg fra mikrokontrollere.

Ferdigheter – kandidaten skal

- Kunne sette opp og installere OS og annen software på Raspberry Pi.

- Kjenne til og bruke grunnleggende elektroniske komponenter som LED, knapper, motstand, buzzer og transistorer.
- Kunne styre eksterne komponenter, med både analoge og digitale signaler, fra embedded maskin (segment display, led matrise, LCD, I/O expansion).
- Kunne lese inn data fra analoge og digitale eksterne sensorer. (Eks: lys, temperatur, knapper.)
- Kunne styre motorer. (Servo, DC, stepper, solenoid, m.m.)
- Kommunisere trådløst med IR, RF og WiFi.
- Bruke data fra eksterne sensorer til å styre eksterne komponenter.
- Kjenne til og bruke viktige protokoller i embedded systems som SPI, I2C, 1Wire og UART.
- Forstå hvordan embedded systems virker i grensesnittet mellom elektronikk og program (eeprom, interrupts, minnehåndtering).
- Kunne sette opp og installere OS og annen software på Raspberry Pi.
- Kunne designe enkle kretskort og dokumentere oppkobling med Fritzing.

Generell kompetanse – kandidaten skal

- Ha innsikt i bruksområder og praktiske anvendelser for embedded systems.
- Designe, teste og kritisk evaluere embedded systems som løsninger på virkelige problemstillinger, som å lage roboter og spillkonsoller.

4.7 PRO300 Virtual Reality prosjekt

Engelsk navn:	Virtual Reality Project
Studiepoeng:	7,5
Fagområde:	Teknologi / IT
Undervisningsspråk:	NO
Program:	Valgemne ved Bachelor i IT – Programmering, Bachelor i IT – Spillprogrammering, Bachelor i IT – Interaktivt design, Bachelor i IT – Intelligente systemer, Bachelor i IT – Frontend og mobilutvikling
Absolutte krav til forkunnskaper:	Ingen
Anbefalte forkunnskaper:	
Undervisning:	Høst
Emneansvarlig:	Kim Baumann Larsen

Innhold

Virtual reality prosjekt er ett tverrfaglig emne der studentene deltar med sin fagkompetanse. Prosjektet har til hensikt å utvikle interaktive virtual reality applikasjoner for spill, kunst eller visualisering. Studentene skal kunne lage konsepter, designe og lage interaktive Virtual Reality løsninger og forstå de tekniske rammer i produksjon av disse.

Læringsutbytte

Kunnskaper – kandidaten skal

- kjenne til ulike maskin- og programvareplattformer for Virtual Reality
- kjenne til ulike plattformkrav til forskjellige Virtual Reality løsninger.
- kjenne til mulighetene og begrensningene med Virtual Reality
- ha forståelse for hva som skal til for å oppnå størst mulig tilstedeværelse i en Virtual Reality løsning
- kjenne til grunnleggende prinsipper for historiefortelling for interaktiv VR

Ferdigheter – kandidaten skal

- kunne skissere og planlegge ett interaktivt Virtual Reality konsept for spill, kunst eller visualisering
- kunne lage prototyper for ulike plattformer
- kunne designe og implementere en interaktiv VR opplevelse for valgt plattform
- kunne integrere 3d modeller, lyd elementer og interaksjonspunkter til en helhetlig VR opplevelse for valgt plattform

Generell kompetanse – kandidaten skal

- kunne vurdere og lage en optimal Virtual Reality løsning for en valgt plattform
- kunne vurdere ulike Virtual Reality plattformer og løsnings muligheter og begrensninger

4.8 ENT200 Entreprenørskap

Engelsk navn:	Effectual Entrepreneurship
Studiepoeng:	7,5
Fagområde:	Ledelse
Undervisningsspråk:	NO
Program:	Obligatorisk ved Bachelor i prosjektledelse kunst og kreativ næring og ved Bachelor i strategi og PR, valgemne ved Bachelor i digital markedsføring
Absolutte krav til forkunnskaper:	Ingen
Anbefalte forkunnskaper:	Ingen
Undervisning:	Høst
Emneansvarlig:	Tor Grønsund

Innhold

Studenten skal lære moderne tilnærminger for å drive frem nye ideer og prosjekter på egen hånd så vel som i etablerte virksomheter. Emnet gir studenten en innføring i hvordan idéer og muligheter kan oppstå, vurderes og utvikles, samt bekjentgjøre seg med etablering, kapitalisering og eget potensiale for nyskapende utviklingsarbeid. Et sentralt tema er forholdet mellom tradisjonelle og entreprenørielle ledelsesperspektiver.

Læringsutbytte

Kunnskaper - etter å ha fullført emnet skal studenten:

- ha grunnleggende kunnskap om moderne forklaringsmodeller og retninger innen entreprenørskap og nyskapende utviklingsarbeid
- kjenne til forholdet mellom tradisjonelle og entreprenørielle ledelsesperspektiver

Ferdigheter - etter å ha fullført emnet skal studenten kunne:

- kunne operasjonalisere relevant teori og rammeverk for å evaluere entreprenørielle prosesser og problemstillinger i næringslivet og samfunnet
- behandle temaene som tas opp i emnet på en selvstendig måte, både muntlig og skriftlig

Generell kompetanse - etter å ha fullført emnet skal studenten kunne:

- ha evne til å kunne kommunisere og formidle faglige synspunkter og erfaringer med andre som har bakgrunn og kunnskap innen entreprenørskapsteori og praksis og gjennom dette bidra til god praksis
- kunne reflektere og kritisk evaluere synspunkter og problemstillinger knyttet til entreprenørskap og entreprenørens rolle i samfunnet