FDM3507 Programmering som designpraktik 7,5 hp

Programming as Design Practice

När kurs inte längre ges har student möjlighet att examineras under ytterligare två läsår.

Fastställande

Kursplan för FDM3507 gäller från och med VT15

Betygsskala

Utbildningsnivå

Forskarnivå

Särskild behörighet

Alla doktorander eller forskare, med antingen datavetenskaplig/teknisk bakgrund, eller bakgrund i konst och design.

Undervisningsspråk

Undervisningsspråk anges i kurstillfällesinformationen i kurs- och programkatalogen.

Lärandemål

Efter genomförd kurs ska den forskarstuderande kunna:

Använda ett antal olika programmeringsparadigm i små projekt, och argumentera för när varje paradigm är lämpligt.

Använda ett antal olika programmeringspraktiker i små projekt, och argumentera för när varje praktik är lämplig.

Förstå och tillämpa teknologier för interaktiva gränssnitt, samt grunderna i ljud, video, och generativ datorgrafik.

Argumentera kring när det är lämpligt att välja programmering som tillvägagångssätt, och när återanvändning av befintliga lösningar kan vara lämpligare.

Kombinera flera system, befintliga samt nya, för att på så vis skapa en ny entitet som når uppställt designmål.

För detta, skall de lära sig teknologier för interkommunikation mellan program och andra involverade system, så att dessa kan dela kontroll-data, ljudströmmar och videoströmmar sinsemellan.

Kursinnehåll

Traditionellt har målet för programvaruutveckling varit att skapa en förutsägbar, rigorös, transparent process så att en för uppgiften välfungerande mjukvara kan levereras i tid, och inom budget. Datorer används idag i många andra platser än tydligt uppgiftsorientade sammanhang ex in i klassrum, hem, arkitekt-studios, design-studios, och konstnärsateljéer, så har den traditionella arbetsformen för programvaruutveckling förändrats liksom verktygen för det. Nya tillvägagångssätt har uppstått, som går utöver den formella, ingenjörsmässiga synen på programmering. Särskilt påtagligt har de estetiska aspekterna betonats och nya former för hur man organiserar programutvecklingsprojekt uppstått. Kurser kommer särskilt fokusera frågan: På vilket vis kan programmeringsförfarandet främja estetisk kreativitet?

Målet med denna kurs är att lära studenter färdigheter som krävs för programmering inom kreativ praktik. Kursen kommer dels att presentera programmeringsspråk som används i kreativa sammanhang och dels hur man kan närma sig programmering på olika sätt. Kursen kommer att varva exempel på hur utvecklare har använt sig av programmeringsspråken med konkreta färdighetsövningar där studenten använder sig av programmeringsspråken.

Kursupplägg

Disposition

Kursen består av två distinkta block:

1. Teoretiska Föreläsningar
2. Projektarbeten

Föreläsningar och workshops

Under första blocket ges 10 2-timmars föreläsningar.

1. Kreativ programmering: exempel och historia (En föreläsning)

2. Processing-språket (Tre föreläsningar)

a. Grunden

b. Programmering med metoder

c. Objektorientering

3. Pure Data språket (Två föreläsningar)

a. Manipulering av kontroll-data

b. Ljudmanipulering

4. Parameter-mappning, och Open Sound Control – protokollet (En föreläsning)

5. Sensor-plattformar (En föreläsning)

6. Tillvalsföreläsning 1 – innehållet bestäms efter första mötet med deltagarna

7. Tillvalsföreläsning 2 – innehållet bestäms efter första mötet med deltagarna

Denna kurs är kollaborativ: kursdeltagarna turas om att förbereda och presentera föreläsningarna, och att organisera de praktiska studiegrupperna.

På första mötestillfället så berättar alla deltagare om vilka delar i kursen de känner att de vill delta genom att presentera, samt rösta på vad de vill ska vara med i kursen,

Under kursens gång kommer deltagare, individuellt och i par, arbeta med projekt där de tillämpar vad de lärt.

Mellan föreläsningarna, så planeras 10 2-timmars laborationsmöten, där deltagarna tillämpar det de lärt på föregående föreläsning genomsmå projektarbeten. I vilken utsträckning detta efterföljs beror på om kursen har tillräckligt många deltagare som vill vara med och hjälpa till i att hålla i sessionerna.

Projektarbeten

Kursen innefattar tre projektarbeten:

Två små, individuella där studenten provar ett nytt programmeringsspråk, och implementeringsparadigm. Åtminstone det ena måste producera resultat i realtid.

Tredje projektarbetet utförs i par, där två eller fler av tidigare arbeten anpassas så att de kan inkorporeras i en ny kombination, där de tillsammans utgör en ny helhet.

I detta grupparbete försöker vi igen att blanda färdigheter så mycket som möjligt, så att varje tekniskt kunnig deltagare arbetar med en konstnär/designer.

Utöver föreläsningar, ges även mycket utrymme för reflektion kring praktiken.

Alla dessa projektarbeten utförs under icke-schemalagd tid, som egenarbete.

Under den perioden ska deltagarnaplanera sex 2-timmars möten där man presenterar sina projekt, och deltar i crit-sessioner kring varandras arbeten.

Sex möten kring projektaretet:

1 Ett för att komma igång, och presentera idéer för första projektet.

2 Ett för att visa utvecklingen, och få återkoppling på första projektet (Crit session).

3 Ett för att visa resultat från första projektarbetet, och även presentera idéer för andra projektet.

4 Ett för att visa utvecklingen, och få återkoppling på andra projektet (Crit session).

5 Ett för att visa resultat från andra projektarbetet, skapa grupper för tredje projektet, och börja ta fram idéer för tredje projektet.

6 Ett för att visa framsteg och få återkoppling på tredje projektarbetet (Crit session).

Blogg

Under hela kursens gång, skall varje student skriva på en blogg om sina aktiviteter. Efter varje föreläsning, skall bloggen uppdateras med en personlig reflektion, där föreläsningens innehåll och forskningsartiklar relateras till den egna forskningen.

Målsättningen är att ge en bra grund för att skriva en forskningsartikel, som bygger på erfarenheten med denna kurs. Att skriva en artikel är dock inte ett examinationskrav.

Om studenten har tekniska frågor så skall dessa ställas på relevanta internet-forum. En länk till frågan skall sedan läggas på bloggen, så att ansvarig föreläsare kan svara på frågan. Detta hjälper studenterna att lära sig formulera sina tekniska frågor väl, och att använda dessa forum för att hitta lösningar på tekniska problem.

Efter varje föreläsning, skall varje student skriva ett mindre program, utefter föreläsningens innehåll. De skall även reflektera över hur föreläsningens innehåll relaterade till erfarenheten av att skriva programmet.

Bloggarna läses av medstudenter, som sedan ger återkoppling:

- Alla studenter läser varandras bloggar inför möten (om bloggarna är för många så får varje student ett mindre antal tilldelat).

- Studenterna skall kommentera på blogginnehållet, och ta upp kommentarerna på nästkommande Crit-session.

- Studenter skall reflektera på den feedback de får från Crit-sessionerna, och dokumentera hur de gjort, på sin blogg.

Kurslitteratur

Ca. 20 forskningsartiklar. Dessa innefattar bland annat:

“Processing: Programming for Designers and Artists”, Casey Reas, Ben Fry, Design Management Review, Vol. 20 No. 1, 2009

“Crafting Code at the Demo-scene”, Hansen, Norgård, Halskov,DIS 2014

"Pure Data: another integrated computer music environment."Proceedings of the Second Intercollege Computer Music Concerts, Miller Puckette, (1996): 37-41.

“Code Bending: A New Creative Coding Practice”, Ilias Bergstrom, and Beau Lotto, Leonardo Journal, MIT Press, February 2015, Vol. 48, No. 1

“Open Sound Control: an enabling technology for musical networking”. Matthew Wright (2005), Organised Sound, 10, pp 193-200. doi:10.1017/S1355771805000932.

“OSC-Namespace and OSC-State: schemata for describing the namespace and state of OSC-enabled systems”, Ilias Bergstrom, Joan Llobera, New Interfaces for Musical Expression, NIME’14, June 30 – July 3, 2014

“Soma: live performance where congruent musical, visual, and proprioceptive stimuli fuse to form a combined aesthetic narrative”, Ilias Bergstrom, Beau Lotto, Leonardo Journal, MIT Press (Accepted for publication, posted online August 26, 2014)

“Mutable mapping: gradual re-routing of OSC control data as a form of artistic performance”, Bergström et al, ACE 2009

“The Practices of Programming”, Ilias Bergström, Kristina Höök, in preparation.

Examination

Examinator beslutar, baserat på rekommendation från KTH:s handläggare av stöd till studenter med funktionsnedsättning, om eventuell anpassad examination för studenter med dokumenterad, varaktig funktionsnedsättning.

Examinator får medge annan examinationsform vid omexamination av enstaka studenter.

Examinationen består av:

1. Ett aktivt deltagande under föreläsningarna och övningssessionerna.

a. Varje deltagare presenterar minst en föreläsning och/eller organiserar en övningssession.

b. Man skall inte missa fler än 2 föreläsningar, 2 övningssessioner, och 2 Crit-sessioner. Om du missar fler, tala då med din kursplanerare om hur du kan kompensera för detta.

2. Alla tre projektarbeten skall ha utförts och presenterats.

3. Hela processen finns dokumenterad av studenten på en individuell blogg. Bloggen räknas som examinationsunderlag, både genom processen den beskriver, och kvaliteten i sig.

Etiskt förhållningssätt

- Vid grupparbete har alla i gruppen ansvar för gruppens arbete.
- Vid examination ska varje student ärligt redovisa hjälp som erhållits och källor som använts.

- Vid muntlig examination ska varje student kunna redogöra för hela uppgiften och hela lösningen.