

## TIM / OBOR / PROGRAM

### NÁZEV PŘEDMĚTU

CZ: Softwarové umění

EN: Software Art

### PŘEDPOKLADY PŘEDMĚTU

- 1) Schopnost číst texty v angličtině
- 2) Vlastní notebook
- 3) Základní znalost práce s PC

### PREREQUISITES AND CO-REQUISITES

- 1) Ability to read texts in English
- 2) Student's own laptop
- 3) Basic knowledge of working with PC

### CÍLE PŘEDMĚTU (VE VYUČOVACÍM JAZYCE), MAX. 3 000 ZNAKŮ.

#### Charakteristika:

Softwarové umění je novým uměleckým žánrem, který se zrodil na konci druhého tisíciletí. Představuje pokus o uchopení softwaru jako uměleckého vyjadřovacího prostředku, který nemusí figurovat pouze jako nástroj k produkci klasických uměleckých artefaktů (obrazu, hudby, videa), ale může se sám o sobě stát uměleckým dílem. Umělci, kteří se věnují tomuto uměleckému žánru, odkrývají kreativní potenciál softwaru, rozvíjí inovativní způsoby jeho užití a vytvářejí alternativní modely softwaru. Tento kurz uvádí studenty do problematiky softwarového umění prostřednictvím teoreticky orientovaných přednášek i praktičtěji zaměřených seminářů.

#### Cíle předmětu:

- 1) v rámci kurzu se studenti obeznámí s definicí softwarového umění z hlediska jeho historického vývoje, estetických preferencí, významných umělců a uměleckých skupin
- 2) obeznámení studentů s genealogií softwarového umění
- 3) studenti se seznámí s klíčovými texty softwarového umění
- 4) přiblížení kreativních praxí, které využívají SW umělci, a to formou praktických úkolů
- 5) pochopení pozice softwarového umění v rámci širšího kontextu digitálního umění

## MAIN OBJECTIVES:

### Course characteristics:

Software art is a new artistic genre that was born at the end of the second millennium. It represents an attempt to grasp the software as an artistic means of expression, which may not be merely a tool for producing classical artistic artifacts (image, music, video), but can become an artistic work itself. Artists devoted to this artistic genre reveal the creative potential of software, develop innovative ways to use it, and create alternative software models. This course introduces students to the subject of software art through theoretically oriented lectures as well as more practical seminars.

### Course objectives:

- 1) during the course the students acquaint themselves with the definition of software art in terms of its historical development, aesthetic preferences, prominent artists and artistic groups
- 2) acquaint students with the genealogy of software art
- 3) students will get acquainted with the key texts of the subject of software art
- 4) provide an insight into creative practice used by SW artists in the form of practical tasks
- 5) understand the position of software art within the wider context of digital art

### VÝSTUPY Z UČENÍ:

- 1) poučeně diskutovat o historických aspektech softwarového umění a jeho vztahu k ostatním uměleckým tendencím
- 2) identifikovat a shrnout základní rysy softwarového umění
- 3) schopnost analyzovat softwarové dílo z různých metodologických hledisek
- 4) obeznámení studentů s programováním jako kreativní praxí (programování jako objevování)
- 5) seznámení s anatomíí softwarových děl

### LEARNING OUTCOMES:

- 1) ability to discuss the historical aspects of software art and its relation to other artistic tendencies on a specialistic level
- 2) identify and summarize the basic features of software art
- 3) ability to analyze a work of software art from different methodological perspectives
- 4) familiarize students with programming as creative practice (programming as discovery)
- 5) familiarize students with the anatomy of software art works

## ČASOVÁ OSNOVA:

1. **21.9** Úvodní hodina: Obsah předmětu, požadavky na ukončení
2. **28.9 Výuka se nekoná.**
3. **5.10** Home-read  
Texty: Tools of Distorted Creativity  
<https://transmediale.de/content/tools-distorted-creativity>  
Interview with Adrian Ward (s. 114-120)  
<https://i-dat.org/wp-content/blogs.dir/28/files/2019/11/Users-Guide.pdf>
4. **12.10** Ukázky softwarových děl a jejich interpretace
5. **19.10** Historie softwarového umění 1: Konceptuální východiska softwarového umění – konceptuální umění, umění a technika, net.art - neprobíhá online (video přednáška)
6. **26.10** Historie softwarového umění 2: Zrod softwarového umění
7. **2.11** – Kreativní apropiace webové stránky
8. **9.11** Kreativní programování 1: úvod
9. **16. 11** Kreativní programování 2: práce s textem
10. **23.11** Kreativní programování 3: generování obrazu
11. **30.11** Software art – poezie – Florian Cramer, Code works  
Aktivita: tvorba codeworks
12. **7.12** Destruktivní kreativita
13. **14.12** Softwarové umění v galerijní praxi

## COURSE CONTENT

- 1) Introduction: Course content, course completion requirements
- 2) The formation of software art from net.art (lecture and discussion)
- 3) Seminar - Creative programming
- 4) Conceptual shift of software art - conceptual art, art and technology, net.art (lecture and discussion)
- 5) Seminar - Creative programming
- 6) Invited guest – artist
- 7) Seminar - Creative programming
- 8) Demonstration and interpretation of work of software art (lecture and discussion)
- 9) Seminar - Creative Programming
- 10) Destructive creativity (lecture and discussion)

- 11) Seminar - Creative Programming
- 12) Credit Week: presentations of created works and discussion

## LITERATURA

### Povinná

ARNS, Inke. *Read\_me, run\_me, execute\_me. Code as Executable Text: Software Art and its Focus on Program Code as Performative Text.* in: Rudolf Frieling / Dieter Daniels (Hg.), *Medien Kunst Netz 2: Thematische Schwerpunkte*, Springer Wien/New York 2005, ISBN: 3211238719, S. 177-193 (dt.), S. 197-207 (engl.). Dostupné on-line: [http://www.mediaartnet.org/themes/generative-tools/read\\_me/scroll/](http://www.mediaartnet.org/themes/generative-tools/read_me/scroll/)

BROECKMANN, Andreas. *Software Art Aesthetics*. Mono, No. 1, July 2007, Porto: FBAUP, s. 158-167 Dostupné on-line: <http://www.luisfilipeteixeira.com/fileManager/file/BroeckmannSoftwareArt.pdf>

COX, Geoff – McLEAN, Alex – WARD, Adrian. *The Aesthetics of Generative Code*. [online]. Paper delivered at the Generative Art 00 conference, Milan, 2000. [cit. 23. 10. 2015]. Dostupné z: <http://generative.net/papers/aesthetics/>.

CRAMER, Florian – GABRIEL, Ulrike. *Software Art*. in Andreas Broeckmann, Susanne Jaschko (eds.), *DIY Media - Kunst und digitale Medien: Software - Partizipation - Distribution*. [online]. 2001. *Transmediale.01*, Berlin, 2001, s. 29-33. [cit. 3. 1. 2015]. Dostupné z: [http://www.netzliteratur.net/cramer/software\\_art\\_-\\_transmediale.html](http://www.netzliteratur.net/cramer/software_art_-_transmediale.html).

### Doporučená

BOOKCHIN, Natalie – SHULGIN, Alexei. *Introduction to net.art (1994-1999)*. [online]. 1999. [cit. 23. 11. 2015]. Dostupné z: <http://www.easylife.org/netart/>.

BROECKMANN, Andreas. *Net.Art, Machines, and Parasites*. [online]. 1997. [cit. 22. 11. 2015]. Dostupné z: <http://www.nettime.org/Lists-Archives/nettime-l-9703/msg00038.html>.

BURNHAM, Jack. *Software - Information Technology: Its New Meaning for Art*. Exhibition Catalogue. New York: The Jewish Museum, 1970. Catalog Card No.: 70-130192.

CRAMER, Florian. *Concepts, Notations, Software, Art*. [online]. 2002. [cit. 3. 1. 2015]. Dostupné z: [http://www.netzliteratur.net/cramer/concepts\\_notations\\_software\\_art.html](http://www.netzliteratur.net/cramer/concepts_notations_software_art.html).

CRAMER, Florian. *Entering the Machine and Leaving It Again: Poetics of Software in Contemporary Art*. Essay/Lecture, Milano, 2006.

CRAMER, Florian. *Words Made Flash. Code, Culture, Imagination*. Media Design Research, Rotterdam: Piet Zwart Institute, 2005, 140. s.

GORIUNOVA, Olga. *Art Platforms and Cultural Production on the Internet*. New York: Routledge, 2012, 112 s. ISBN 978-0-415-89310-7

GORIUNOVA, Olga – SHULGIN, Alexei. *Glitch*. In FULLER, Matthew (ed.). *Software Studies/ A Lexicon*. Cambridge, Massachusetts: MIT Press, 2008, 334 s. ISBN 978-0-262-06274-9. s. 111–114

FULLER, Matthew (ed.). *Software Studies/ A Lexicon*. Cambridge, Massachusetts: MIT Press, 2008, 334 s. ISBN 978-0-262-06274-9

YOUNG, La Monte – LOW, Jackson Mac. *An Anthology of Chance Operations*. New York: Published by La Monte Young and Jackson Mac Low, 1963, 120 s.

#### RECOMMENDED OR REQUIRED READING

ARNS, Inke. *Read\_me, run\_me, execute\_me. Code as Executable Text: Software Art and its Focus on Program Code as Performative Text*. in: Rudolf Frieling / Dieter Daniels (Hg.), *Medien Kunst Netz 2: Thematische Schwerpunkte*, Springer Wien/New York 2005, ISBN: 3211238719, S. 177-193 (dt.), S. 197-207 (engl.). Dostupné on-line: [http://www.mediaartnet.org/themes/generative-tools/read\\_me/scroll/](http://www.mediaartnet.org/themes/generative-tools/read_me/scroll/)

BOOKCHIN, Natalie – SHULGIN, Alexei. *Introduction to net.art (1994-1999)*. [online]. 1999. [cit. 23. 11. 2015]. Dostupné z: <http://www.easylife.org/netart/>.

BROECKMANN, Andreas. *Net.Art, Machines, and Parasites*. [online]. 1997. [cit. 22. 11. 2015]. Dostupné z: <http://www.nettime.org/Lists-Archives/nettime-l-9703/msg00038.html>.

BROECKMANN, Andreas. *Software Art Aesthetics*. Mono, No. 1, July 2007, Porto: FBAUP, s. 158-167

BURNHAM, Jack. *Software - Information Technology: Its New Meaning for Art*. Exhibition Catalogue. New York: The Jewish Museum, 1970. Catalog Card No.: 70-130192.

COX, Geoff – McLEAN, Alex – WARD, Adrian. *The Aesthetics of Generative Code*. [online]. Paper delivered at the Generative Art 00 conference, Milan, 2000. [cit. 23. 10. 2015]. Dostupné z: <http://generative.net/papers/aesthetics/>.

CRAMER, Florian. *Concepts, Notations, Software, Art*. [online]. 2002. [cit. 3. 1. 2015]. Dostupné z: [http://www.netzliteratur.net/cramer/concepts\\_notations\\_software\\_art.html](http://www.netzliteratur.net/cramer/concepts_notations_software_art.html).

CRAMER, Florian. *Entering the Machine and Leaving It Again: Poetics of Software in Contemporary Art*. Essay/Lecture, Milano, 2006.

CRAMER, Florian – GABRIEL, Ulrike. *Software Art*. in Andreas Broeckmann, Susanne Jaschko (eds.), *DIY Media - Kunst und digitale Medien: Software - Partizipation - Distribution*. [online]. 2001. *Transmediale.01*, Berlin, 2001, s. 29-33. [cit. 3. 1. 2015]. Dostupné z: [http://www.netzliteratur.net/cramer/software\\_art\\_-\\_transmediale.html](http://www.netzliteratur.net/cramer/software_art_-_transmediale.html).

CRAMER, Florian. *Words Made Flash. Code, Culture, Imagination*. Media Design Research, Rotterdam: Piet Zwart Institute, 2005, 140. s.

GORIUNOVA, Olga. *Art Platforms and Cultural Production on the Internet*. New York: Routledge, 2012, 112 s. ISBN 978-0-415-89310-7

FULLER, Matthew (ed.). *Software Studies/ A Lexicon*. Cambridge, Massachusetts: MIT Press, 2008, 334 s. ISBN 978-0-262-06274-9

YOUNG, La Monte – LOW, Jackson Mac. *An Anthology of Chance Operations*. New York: Published by La Monte Young and Jackson Mac Low, 1963, 120 s.

PŘEDPOKLÁDANÉ VÝUKOVÉ ČINNOSTI A VÝUKOVÉ METODY  
přednášky, diskuze a praktická cvičení

PLANNED LEARNING ACTIVITIES AND TEACHING METHODS  
lectures, discussions and practical exercises

### **METODY A KRITÉRIA HODNOCENÍ**

- 1) Vytvoření softwarového díla (webová stránka, program vytvořený v jazyce Python) doplněného o jeho kritickou interpretaci (teoretický text s rozsahem 1800 znaků včetně mezer)
- 2) Vytvoření návrhu imaginárního programu, který komentuje stereotypy spjaté s masově využívanými softwarovými aplikacemi nebo se vyjadřuje k rozmanitým ekologickým, politickým, sociálním, etickým či uměleckým otázkám, případně navrhuje způsoby jejich řešení. Návrh zahrnuje funkce programu, jeho vizuální podobu a autorský záměr (1800 znaků)
- 3) Minimální docházka v tomto předmětu je stanovena na 75 procent

**Úkoly je nutné odevzdat do 10.1.2022 do příslušné odevzdávací skříně.**

### **ASSESSMENT METHODS AND CRITERIA**

- 1) Creation of an artwork supplemented by its critical interpretation (a theoretical text with a minimum of 1800 characters including spaces)
- 2) Creation of a plan of imaginary program. The plan has to include functions of the imaginary software, its visual design, and author intent (1800 characters including spaces)