

# **Coding Tools for IM**

# **CODING TOOLS**

**PRIMARY CREATIVE AND EXPRESSIVE MEANS IN A DIGITAL CULTURE**

[CHARLIE GERE, *DIGITAL CULTURE*, 2005 (IN READING LIST, FURTHER READING)]

- OFFER HUGE POTENTIAL FOR ARTISTIC EXPLORATION AND EXPRESSION

- REQUIRE PROCEDURAL FLUENCY

(ALGORITHMIC THINKING AND CONVERTING EVENTS INTO LANGUAGE)

# **KEY ASPECTS OF CREATIVE APPROACH TO CODING**

- **UNDERSTANDING BASIC PRINCIPLES OF IT, COMPUTING AND PROGRAMMING**
  - **SELECTING BEST TOOLS FOR A SPECIFIC CONTEXT**
    - **LEARNING SEVERAL PL (IN PARALLEL)**
      - **READINESS TO LEARN NEW**

# **AROUND 700 PROGRAMMING LANGUAGES**

## **SOME POPULAR WITH THE ARTISTS**

**GAME DEVELOPMENT ENGINES: UNREAL, UNITY, GODOT**

**VISUAL PL: MAX/MSP, PURE DATA, VVVV, TOUCHDESIGNER**

**LIBRARIES: PROCESSING, P5.JS, OPENFRAMEWORKS, FFMPEG**

**PL: SUPERCOLLIDER, PYTHON, C++**

# **GAME DEVELOPMENT ENGINES**

**UNREAL**

**WRITTEN IN C++**

**POWERFUL, PORTABILITY, LOTS OF ASSETS**

**LIMITATIONS:**

**COMMERCIAL, STEEP LEARNING CURVE**

**UNITY**

**WRITTEN IN C++**

**C# SCRIPTING, POWERFUL, PORTABILITY, LOTS OF ASSETS**

**LIMITATIONS:**

**COMMERCIAL, STEEP LEARNING CURVE**

**GODOT**

**WRITTEN IN C++**

**PROGRAMMING/SCRIPTING IN C#, GDSCRIPT, VISUALSCRIPT, PYTHON,**

**POWERFUL, PORTABILITY, OPEN SOURCE, EASIER TO LEARN,**

**SOLID LEARNING RESOURCES**

**LIMITATIONS:**

**ASSETS AND LEARNING RESOURCES COMPARED TO COMMERCIAL**



# **VISUAL PROGRAMMING LANGUAGES**

**MAX/MSP/JITTER**

**WRITTEN IN C AND C++**

**INTERACTIVE PROGRAMMING FOR MUSIC AND MULTIMEDIA**

**POWERFUL, LARGE USER AND DEVELOPER BASE, GOOD LEARNING RESOURCES**

**LIMITATIONS:**

**REQUIRES MUSICAL KNOWLEDGE,**

**VISUAL INTERFACE NOT SUITABLE FOR COMPLEX PROJECTS,**

**PRAGMATICISM OBSCURES UNDERSTANDING,**

**PROPRIETARY**

**PURE DATA**

**WRITTEN IN C**

**INTERACTIVE PROGRAMMING FOR MUSIC AND MULTIMEDIA**

**POWERFUL, LARGE USER AND DEVELOPER BASE, GOOD LEARNING RESOURCES**

**OPEN SOURCE**

**LIMITATIONS:**

**QUIRKY/IDIOSYNCRATIC, REQUIRES MUSICAL KNOWLEDGE,  
VISUAL INTERFACE NOT SUITABLE FOR COMPLEX PROJECTS,**

**PRAGMATICISM OBSCURES UNDERSTANDING**

**WWW**

**WRITTEN IN DELPHI, PLUGINS CAN BE DEVELOPED IN .NET AND C#  
MOTION GRAPHICS, A/V SYNTHESIS IN LARGE MEDIA ENVIRONMENTS  
WITH PHYSICAL INTERFACES  
POWERFUL, FAST, STRONG GRAPHICS SUPPORT (OGL), OPEN SOURCE**

**LIMITATIONS:**

**QUIRKY/IDIOSYNCRATIC, MODEST USER AND DEVELOPER BASE,  
LACK OF LEARNING RESOURCES,  
VISUAL INTERFACE NOT SUITABLE FOR COMPLEX PROJECTS,  
PRAGMATICISM OBSCURES UNDERSTANDING**

# **TOUCHDESIGNER**

**INITIALLY BASED ON HOUDINI, NOW PYTHON**

**MOTION GRAPHICS, A/V SYNTHESIS WITH PHYSICAL INTERFACES**

**POWERFUL, FAST, STRONG AND FAST GRAPHICS SUPPORT (OPENGL),**

**GOOD LEARNING RESOURCES**

## **LIMITATIONS:**

**UI CAN BE COMPLEX TO LEARN AND MAINTAIN UNDERSTANDING,**

**VISUAL INTERFACE NOT SUITABLE FOR COMPLEX PROJECTS,**

**PRAGMATICISM OBSCURES UNDERSTANDING,**

**PROPRIETARY**

**LIBRARIES**

# **PROCESSING**

**JAVA GRAPHICAL LIBRARY AND PDE**

**RELATIVELY SMOOTH LEARNING CURVE, LOTS OF RESOURCES AND EXAMPLES,**

**MANY PORTS AND SPIN-OFF PROJECTS**

**LIMITATIONS:**

**SPEED, OPAQUENESS,**

**UNCONVENTIONAL REQUIREMENTS LEAD TO JAVA**

**FINDING GOOD LEARNING RESOURCES AND EXAMPLES**

## **P5.JS**

**PROCESSING SYNTAX FOR JAVASCRIPT**

**WORKS IN BROWSER, RELATIVELY SMOOTH LEARNING CURVE,**

**NICE RESOURCES AND EXAMPLES**

**LIMITATIONS:**

**SPEED, UNCONVENTIONAL REQUIREMENTS LEAD TO JAVASCRIPT**



# **OPENFRAMEWORKS**

**C++ GRAPHICAL LIBRARY**

**POWERFUL, FAST, EXTENSIBLE**

**LIMITATIONS:**

**RELATIVELY STEEP LEARNING CURVE,**

**REQUIRES KNOWLEDGE OF C++,**

**LACK OF LEARNING RESOURCES AND EXAMPLES**

# **FFMPEG**

**COMMAND-LINE SUITE OF LIBRARIES AND PROGRAMS**

**FOR HANDLING MULTIMEDIA FILES AND STREAMS**

**POWERFUL, FAST, CAN BE INTEGRATED WITH OTHER PL**

## **LIMITATIONS:**

**COMMAND-LINE TEXT SYNTAX LESS LEGIBLE THAN COMMON PL,**

**ASSETS AND LEARNING RESOURCES TRICKY TO NAVIGATE**

# **PROGRAMMING LANGUAGES**

**SUPERCOLLIDER**

**AVAILABLE IN C++**

**REAL-TIME AUDIO SYNTHESIS, ALGORITHMIC COMPOSITION AND INTERACTION**

**FAST, EXTENSIBLE, POWERFUL, SUPPORTS LIVE CODING,**

**SOLID KNOWLEDGE BASE**

**LIMITATIONS:**

**RELATIVELY STEEP LEARNING CURVE,**

**REQUIRES MUSICAL KNOWLEDGE**

# **PYTHON**

**POWERFUL, FAST, EXTENSIBLE, WELL READABLE**

**LOTS OF PLATFORMS AND IMPLEMENTATIONS**

**LEARNING RESOURCES AND EXAMPLES**

**LIMITATIONS:**

**LEARNING RESOURCES AND EXAMPLES,**

**SPEED IN SOME INSTANCES: INTERPRETED**

**FINDING GOOD LEARNING RESOURCES AND EXAMPLES**

**C++**

**POWERFUL, FAST, HIGHLY MODULAR, EXTENSIBLE  
HUGE NUMBER OF LIBRARIES, PLATFORMS AND IMPLEMENTATIONS,  
LEARNING RESOURCES AND EXAMPLES**

**LIMITATIONS:**

**VASTNESS (NUMBER OF LIBRARIES, MODES AND IMPLEMENTATION SCENARIOS),  
FINDING GOOD LEARNING RESOURCES AND EXAMPLES**

## **ONLINE TOOLS**

**STACK OVERFLOW: Q&A**

**GITHUB: EXTENSIVE SOFTWARE DEVELOPMENT VERSION SYSTEM**

# TAKEAWAYS

- THE EXECUTABLE PROCEDURE OF ANY CREATIVE PROCESS

WHICH CAN BE CLEARLY DEFINED

CAN BE ALGORITHMIZED AND CODED (AND MATERIALIZED DEPENDING ON HARDWARE)

- PLASTICITY AND ADAPTABILITY IN MIMICKING NATURAL PROCESSES

ARE THE DEFINING FACTORS OF UNIVERSAL COMPUTING MACHINE

(THE CONCEPTUAL FOUNDATION FOR MODERN COMPUTER SCIENCE)

- ACHIEVING THAT IS A CREATIVE ENTERPRISE THAT REQUIRES INGENUITY,

INTERDISCIPLINARY RESEARCH, UNDERSTANDING OF ACCUMULATED KNOWLEDGE,

AND CONTINUOUS LEARNING.



## **2 MODES OF THINKING FOR IM PROJECT DEVELOPMENT**

- **MATCHING THE ALGORITHMIC AND THE UNPREDICTABLE ELEMENTS  
INTO A COHERENT SYSTEM**

**(RELIES ON THE ANTICIPATION OF PERFORMATIVE QUALITIES OF THE SYSTEM,  
BASED UPON EXPERIENCE, KNOWLEDGE AND INTUITION)**

- **CONSTRUCTION OF ALGORITHMS AS MULTI-PURPOSE TOOLS  
(REQUIRES PROCEDURAL LITERACY AND PROGRAMMING SKILLS)**

# **3 LAYERS OF PROCEDURAL LITERACY/THINKING**

- **DEMATERIALIZATION OF CERTAIN PHENOMENON INTO A SET OF SIGNS THAT DESCRIBE IT PROPERLY**
  - **RESOLVING THAT SIGN-SET INTO PURE SYNTAX  
(REMOVING THE SEMANTIC LAYER = MEANING)**
- **TRANSLATION OF THE SYNTAX INTO A SERIES OF OPERATIONS  
(WITHIN THE PROGRAMMING ENVIRONMENT)**

# **REQUIRES A SPECTRUM OF COGNITIVE ABILITIES**

- **SENSE FOR RECOGNIZING THE PHENOMENON WHICH CAN BE ALGORITHMIZED  
UNDER GIVEN CONDITIONS**
  - **IMAGINATION AND FLEXIBILITY OF REASONING**
- **DISTINGUISHING BETWEEN THE RATIONAL AND IRRATIONAL ASPECTS IN OUR  
MENTAL CONCEPTS OF NATURAL PHENOMENA**
- **ATTENTION TO THE SCOPE AND LIMITATIONS OF THE ALGORITHMIC SYSTEM.**

# **SYSTEMIC CHALLENGES OF PROCEDURAL THINKING**

- **CONCEPTUAL CONSTRAINTS OF PROGRAMMING LANGUAGES AND HARDWARE ARCHITECTURES IMPOSE CERTAIN SOLUTIONS AND SPIN THE ARTISTIC PROCESS**
- **FIXED PERFORMATIVE CAPABILITIES OF THE HARDWARE CAN REFLECT IN ROUGHNESS AND LACK OF SPONTANEITY**
- **UNDECIDABLE PROBLEMS IN COMPUTABILITY THEORY, AND THE LIMITS OF MATHEMATICAL FORMALIZATION (ESTABLISHED IN GÖDEL'S INCOMPLETENESS THEOREMS)**

