

cs121 - software development introduction

alexandre r.j. françois

visiting associate professor of computer science



outline

course objectives

topics covered

project

interactive systems

examples

project ideas

objective

introduction to
theory and practice
of software design and development...

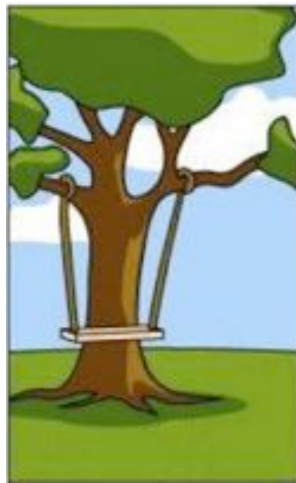
... in a fun and practical way!

why it is worth studying:
software is ubiquitous
software failures can be costly and dangerous
software design and implementation is hard!

challenge



How the customer explained it



How the Project Leader understood it



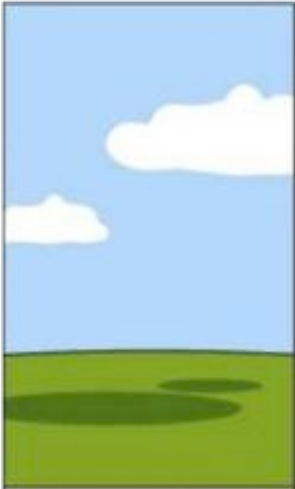
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



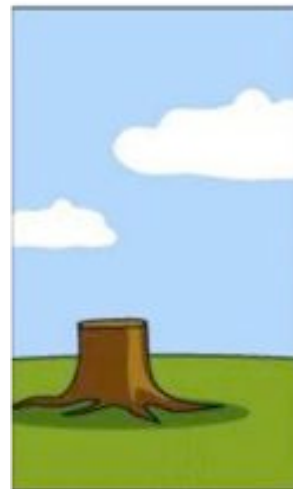
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

key processes of software development

requirements

design

implementation (cs70)

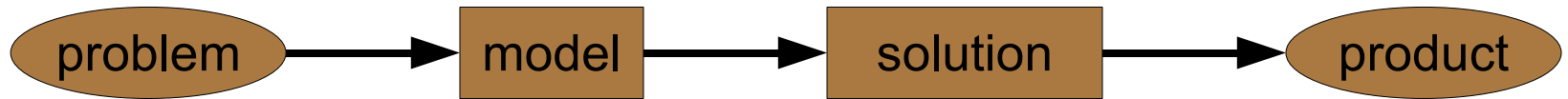
testing

maintenance

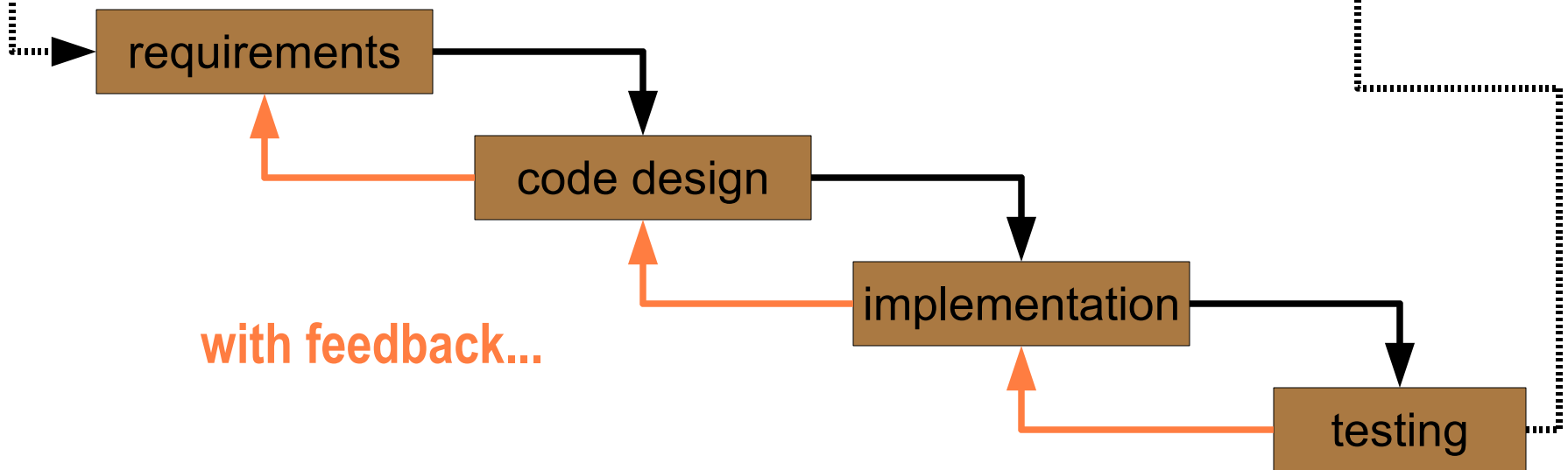
software life cycle: how to organize the key processes of software development

challenge

engineering:



s/w engineering: waterfall model (50s)



waterfall does not work!

initial requirements are speculative

initial design is speculative

speculative decisions compound:

build the wrong thing,

which does not work anyways!

iterative models

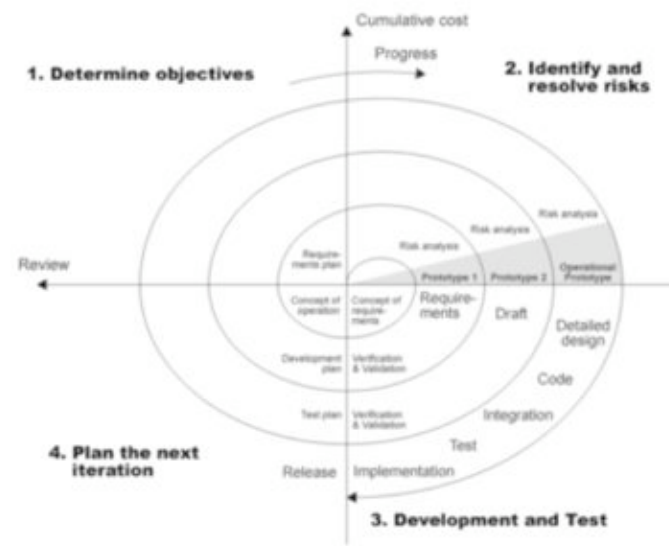
e.g. Boehm's spiral model (1988)

each iteration is like a mini-waterfall

- identify the objectives of the iteration
- design a solution to achieve the objectives
- implement the solution
- test the implementation

other example: Rational Unified Process (IBM)

problem: iterative progress tends to assume monotonic convergence towards solution



agile principles

working software, delivered regularly, is the primary measure of progress

high standards of excellence: test regularly and refactor/redesign when necessary

customer involvement is critical

simplicity: just-in-time design/development

adaptability: embrace change

small, cross-functional, self-organizing teams of professionals

example: SCRUM

at the end of the day...

software projects come in all shapes and sizes (and so do software professionals)

different methods work well for some types of projects and not at all for others

cs121 objectives:

understand the problems

understand various solutions to the problems

practice applying solutions to a particular problem

topics covered

design (for usability)

requirements

architecture (system, components, code)

code design

testing

teamwork, reviews, documentation, etc.

people aspects of software!

project

interactive system

classwide project

YOU (we) define the project

we are our own customers!

methodology

“design” approach

- divergence / convergence phases

- prototyping

- document history (non linear) - portfolio

collaborative enterprise

- synergy

- diversity of expertise and interests

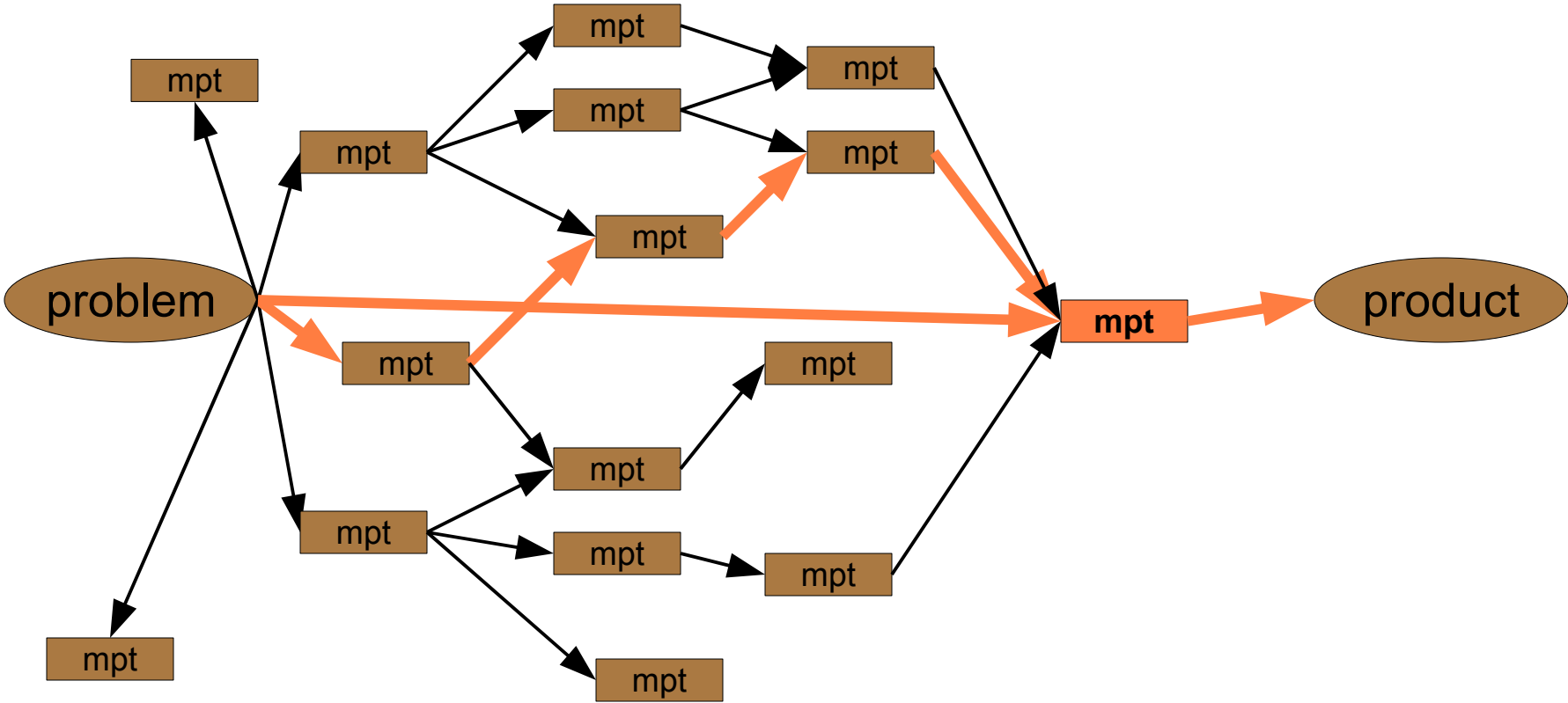
agile iterative approach (with design twist)

- fluid structure

mpt

model | prototype | test

design process



tools – how familiar are you with...

Google Docs

Trac: Wiki, version control, tickets

UML tools(?)

Doxygen

SAI/MFSM

C++

Emacs + gcc?

Xcode?

Google Test?

...

interactive systems

involve **humans**

complex (no matter how complicated...)

situated

embodied

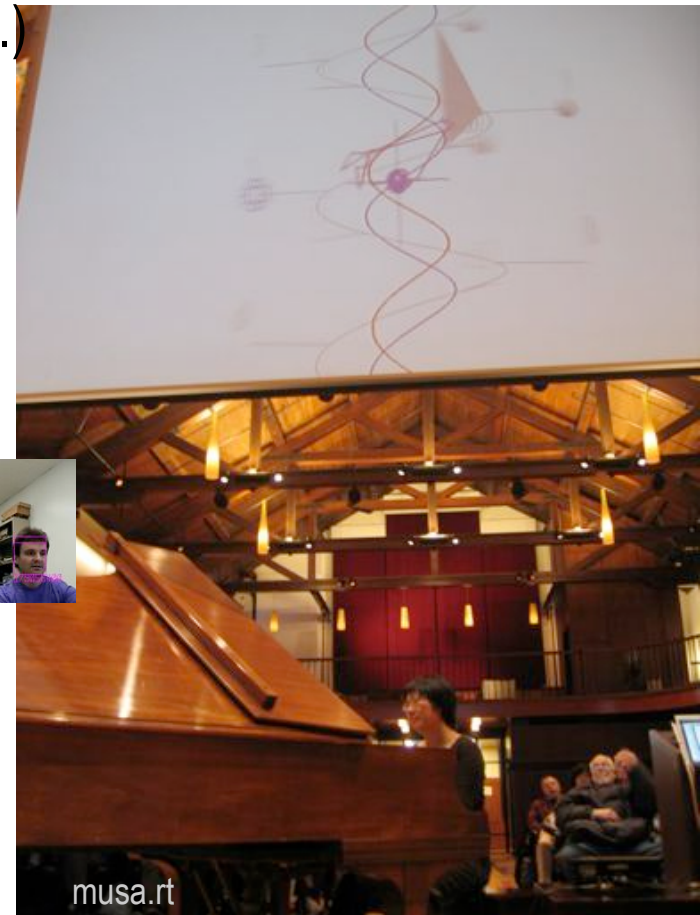
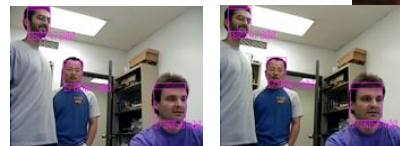
dynamic

desirable **behavior**

robust

adaptive

context-dependent



challenges

by humans for humans

complex task

beyond straightforward computation

system **design**

cross-disciplinary teams

system implementation, testing and maintenance

distributed in time and space

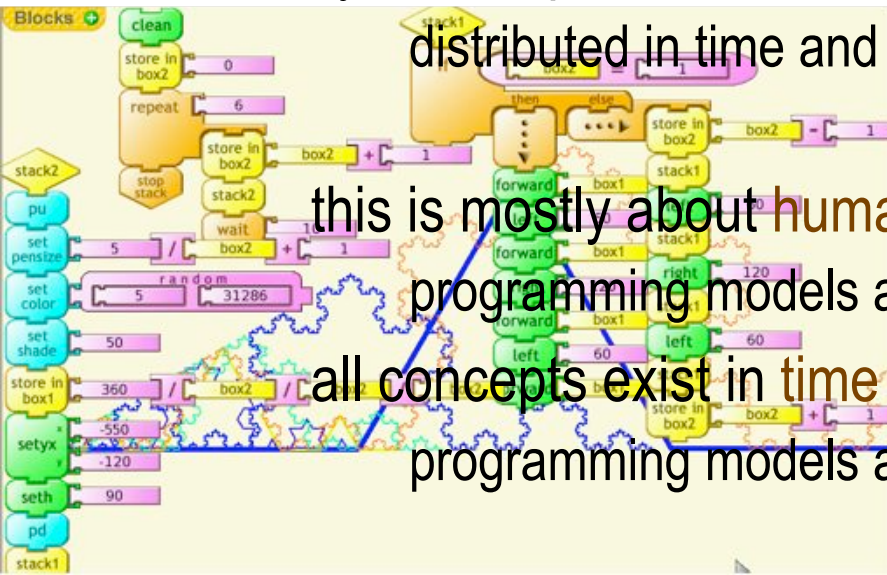
this is mostly about **humans**

programming models and languages are machine-centric

all concepts exist in time

programming models and languages abstract time

turtle art



```
// Character operations (dynamic)
// for all character operations, oracle must be locked
// from calling function if necessary

// iterative learning

void COracle::Reset(){
    a_vecStates.clear();
    a_vecStates.push_back(CState());
    m_nCurrent=0;
    m_nStart=0;
    m_nEnd=0;
}

void COracle::AddCharacter(const CCharacter &c){
    long n=a_vecStates.size();
    a_vecStates.push_back(CState());
    long k=n-1;
    long s;
    while(true){
        long t=a_vecStates[k].FindTransition(c);
        if(t==0)
            break;
        size(
            a_vecStates[k].AddTransition(c,n);
            if(k==0){
                s=k;
                break;
            }
        }
        a_vecStates[k].SetSuffix(s);
    }
    // sticky end
}
```

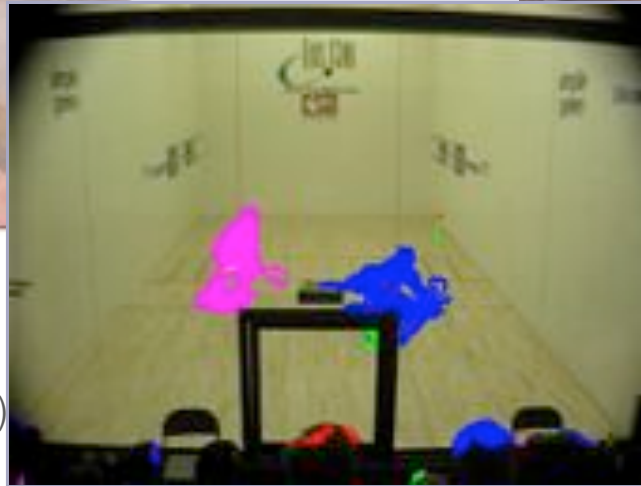
applications -vision-



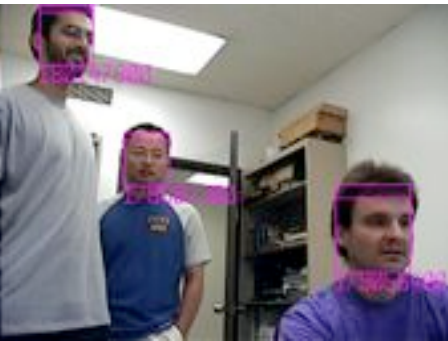
video painting (1999)

[etcv]

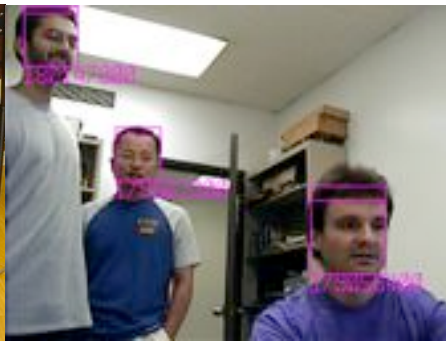
racquetball tracking (2001)



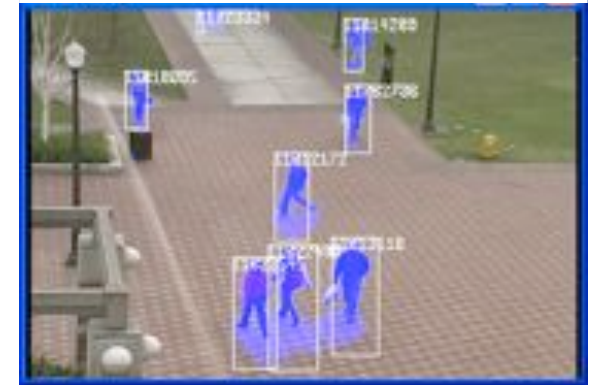
virtual mirror (2000-2002) gra: e.kang
[siggraph2002]



[cviu2007]



stevi (2005-2006) pi: g.medioni, funding: etri



people detection
and tracking (2004)
funding: arda

applications -music-

[nime2006]

[cie2005]

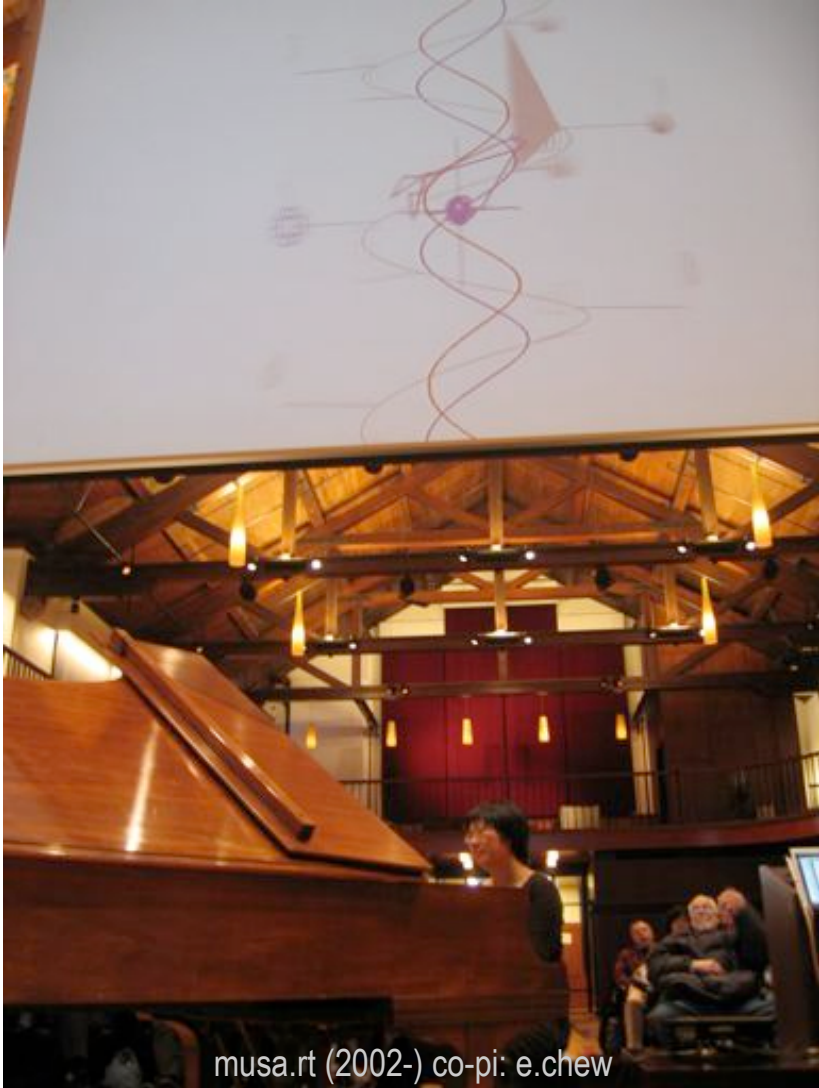
[nime2005]



esp (2004-) co-pi: e.chew, gra: j.liu



music
computation
and cognition
laboratory
at usc



musa.rt (2002-) co-pi: e.chew



mimi (2006-) co-pi: e.chew d.thurmond

[nime2007]

News

Prospective Students | Current Students | Alumni

Home > News & Publications > News > 2007 > Games Students Play, and Make

Games Students Play, and Make



News & Events

News

Select Year: [2009] [2008] [2007] [2006] [2005] [2004]

CS Project Named 1 of 4 Finalists in SIGGRAPH 2009 Research Challenge

June 29, 2009

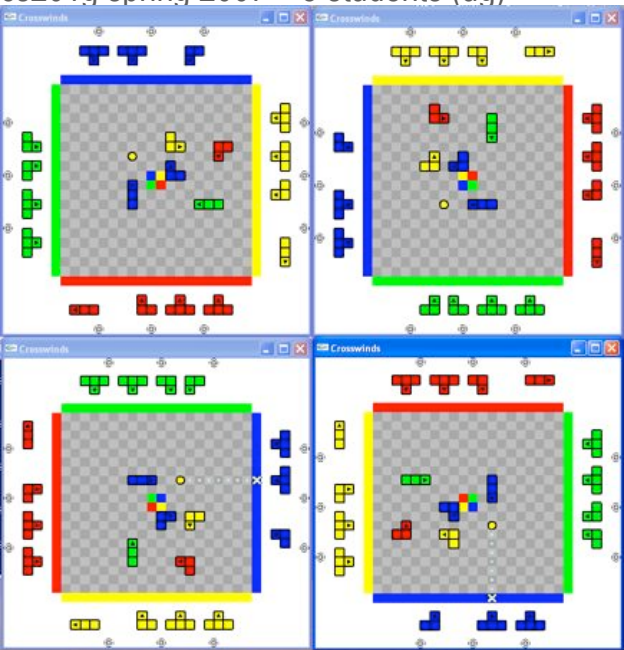
The COMP 150-CIS team project, *An Ant's Life*, will be presented to a panel of distinguished judges in competition for final awards in the SIGGRAPH 2009 Research Challenge competition, during the conference, on August 4, 2009, 1:45-3:00pm, at the Ernest N. Morial Convention Center in New Orleans, LA.

The first person interactive game was collectively designed and prototyped by the 13 students in the Spring session of the course Collaborative Development of Interactive Software Systems (COMP150-CIS), under the guidance of visiting assistant professor Alexandre R.J. François, who created the course.

The SIGGRAPH 2009 competition challenged participants to "choose a specific animal, or a specific animal's sense, and develop a system that will enable a person to experience the physical or social world as that animal does."

In addition, Prof. François's paper about the class-wide project concept was accepted as a poster at ACM Creativity and Cognition 2009.

cs201q spring 2007 – 8 students (ug)



crosswinds

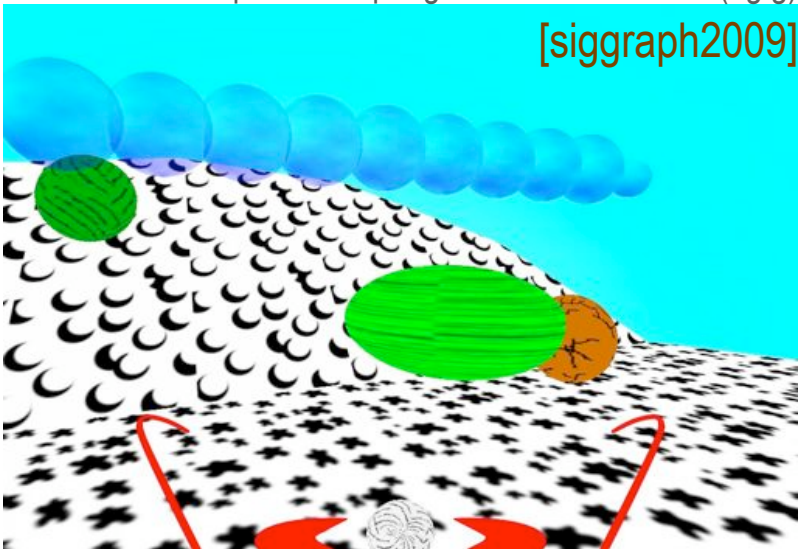
applications
-games-
[c&c2009]

cs599 fall 2002
21 students (g)




comp150-cis spring 2009 – 13 students (ug/g)

[siggraph2009]



an ant's life

interactive games!

interactive systems

- specific requirements

- rich and fun history and industry

- source of innovation

- fun!

evolution of interaction

- input devices and modalities

- graphics

- sound/music

- haptics

project ideas – first round

list here: ...