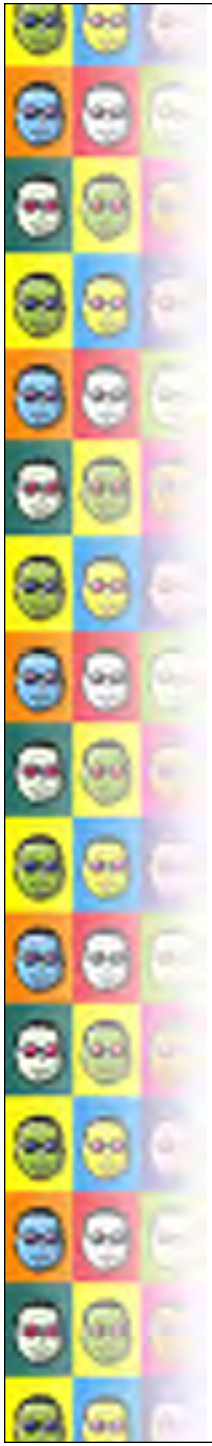


SCALABLE GAME DESIGN

Computing

Computational Thinking

Alexander Repenning, David Webb and Kris Gutierrez
Computer Science & School of Education
University of Colorado

University of Colorado Boulder

Google

Not the usual suspects:
inner city, remote rural,
Native American

Alaska, California,
Wyoming, South Dakota,
Colorado, Texas, ...

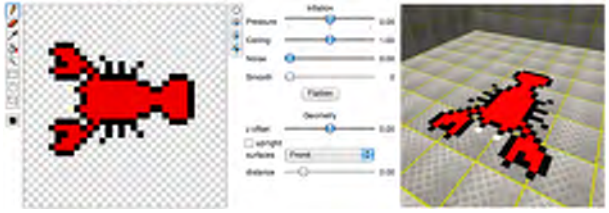
users

projects

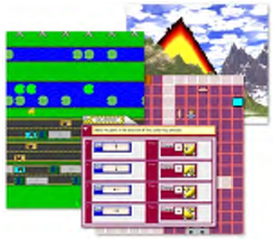
sCALABLE
GAME DESIGN

CS curriculum from elementary
to graduate school

Inflatable Icons

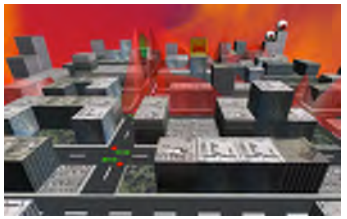


AgentSheets



programming for everyone

AgentCubes



3D simulation authoring
for everyone

Mr Vetro

collective simulations

Computational Thinking Tools

funding

Google

sCALABLE GAME DESIGN

- **goal**: get computer science in to public schools
  - inner city, remote rural, Native American schools
- **approach**:
  1. Reinventing computer science in public schools by motivating & educating all students including women and underrepresented communities to learn about computer science through game design starting at the **middle school level**
  2. Start with **game design**, move on to **science simulation building** and explore **transfer**
  3. Broaden participation through getting game/simulation design into **required courses** if possible (neighborhood: keyboarding, powerpoint)
- **funding**: NSF ITEST + Google, starting: CE21 Type II

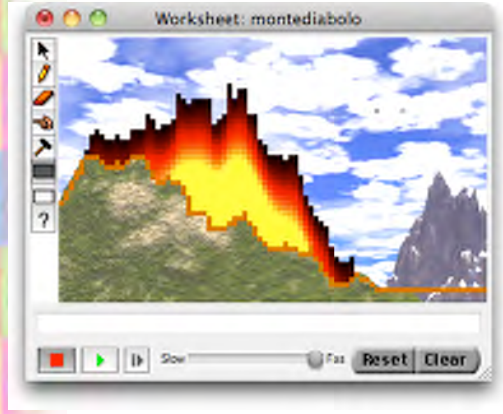
Computational Thinking definition

Good news: we have come a long way

- 2009
  - CT ≠ Programming
  - example: “grandma backing a cake”
- 2011: CSTA, ISTE, NSF: Computational thinking (CT) is a problem-solving process that includes (but is not limited to) the following characteristics:
  - Formulating problems in a way that enables us to use a computer and other tools to help solve them.
  - Logically organizing and analyzing data
  - Representing data through **abstractions** such as **models** and **simulations**
  - Automating solutions through **algorithmic thinking**
  - Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources
  - **Generalizing** and **transferring** this problem solving process to a wide variety of problems

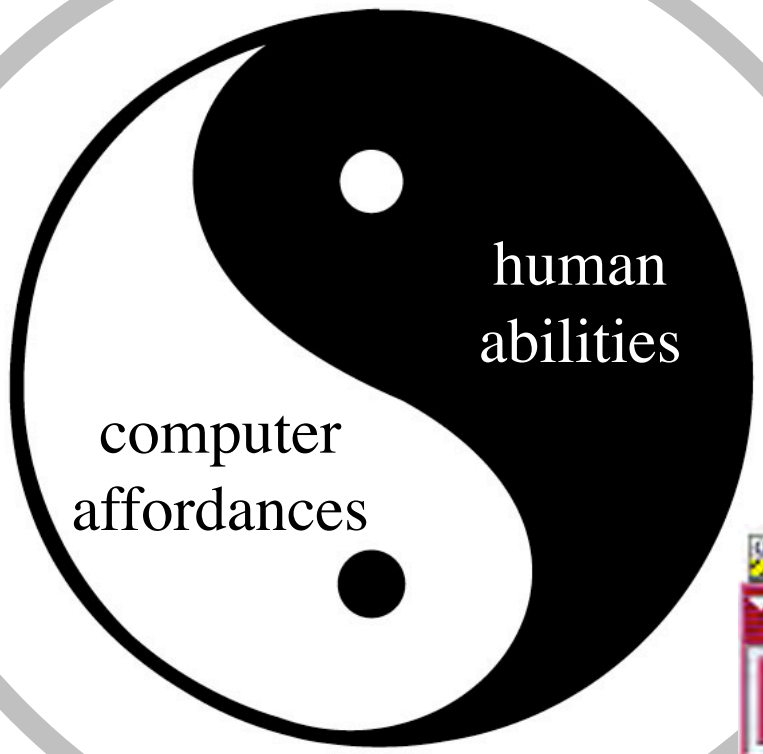
computational thinking tools

synthesize **human abilities** with **computer affordances**



start with question: how does a mudslide work?

revise model

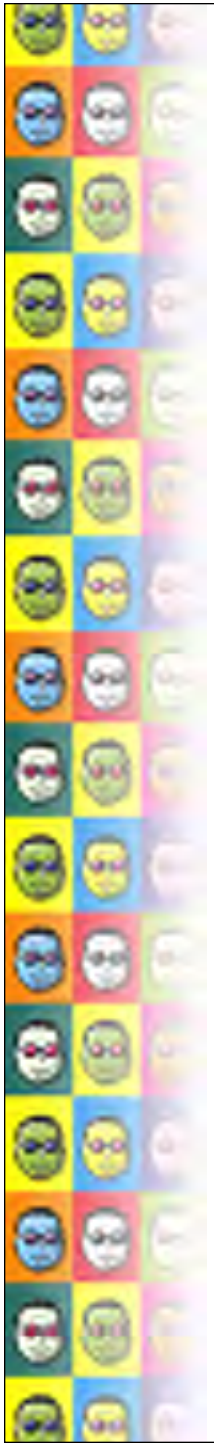


visualize consequence of thinking

express ideas computationally

run model


9NEWS.com
COLORADO'S NEWS LEADER
TEACHERS &
VIDEO GAMES
5:17 81°



results

What kind of pedagogy should be used to broaden participation of women and minorities?

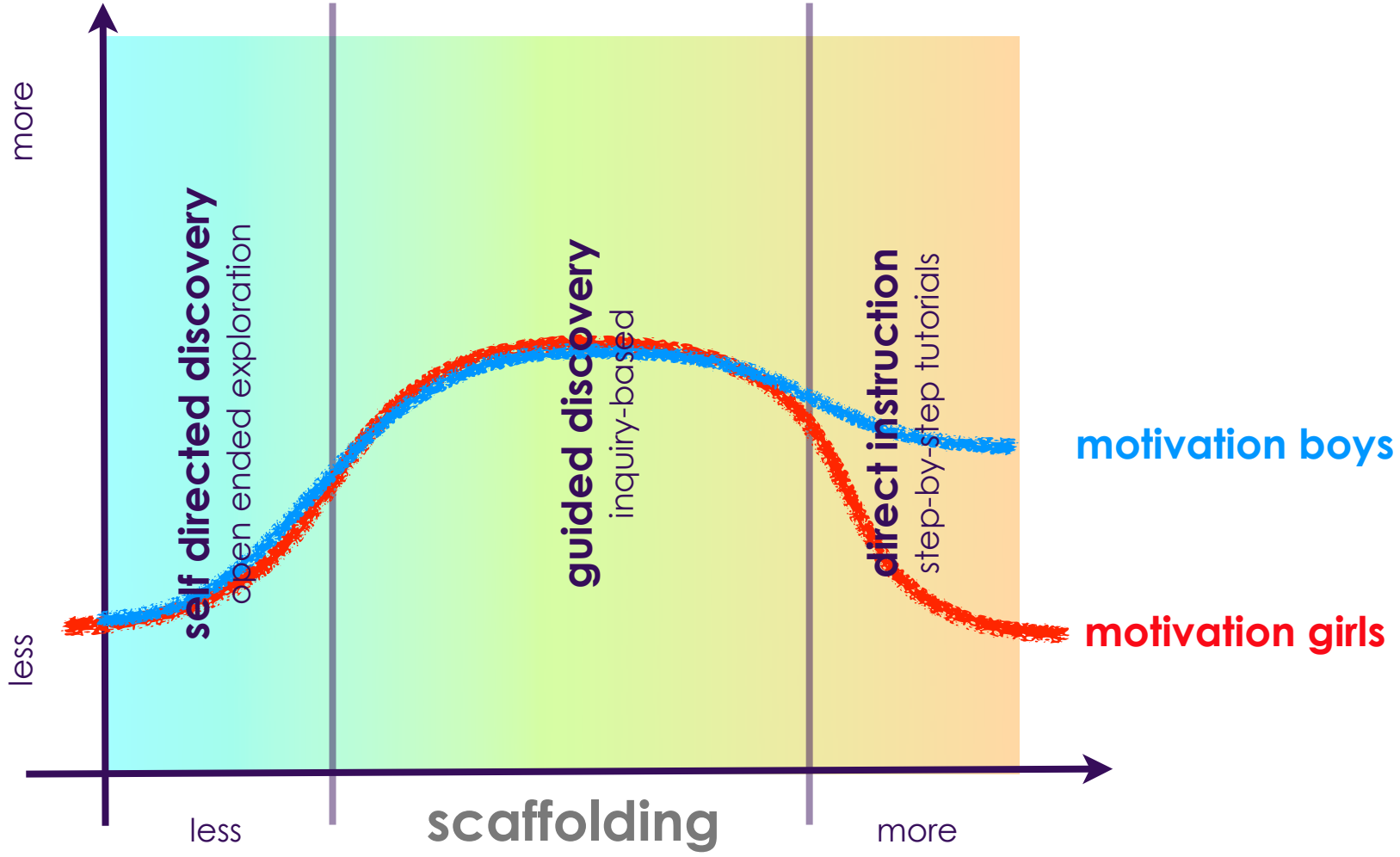
findings

SCALABLE GAME DESIGN

- ***Reach***: already > 4000 students in year 2 of three-year project (Alaska, Colorado, Ohio, Oregon, South Dakota, Tennessee, Texas, and Wyoming)
  - some schools: 600 students/year/school
  - starting internationally, e.g., Brazil site.
- ***Broadening participation***: 45% girls, 55% boys; 44% white, 56% racial minorities
- ***Motivation***: 61% of the girls, 71% of boys; 71% of white students, 69% racial minority students want to continue with similar courses
- ***Learning outcomes***: Computational Thinking Pattern Analysis: every game submitted (one every 14 seconds during class) gets analyzed

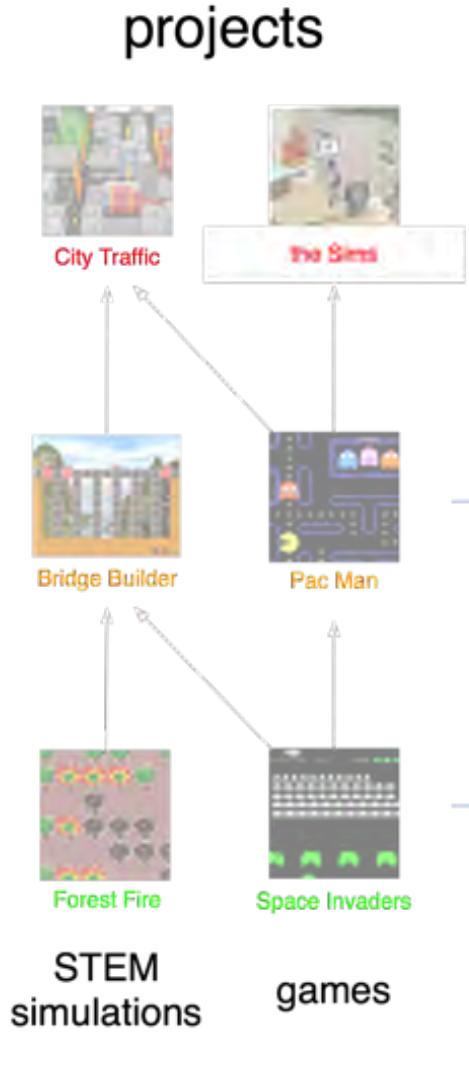
motivation versus scaffolding

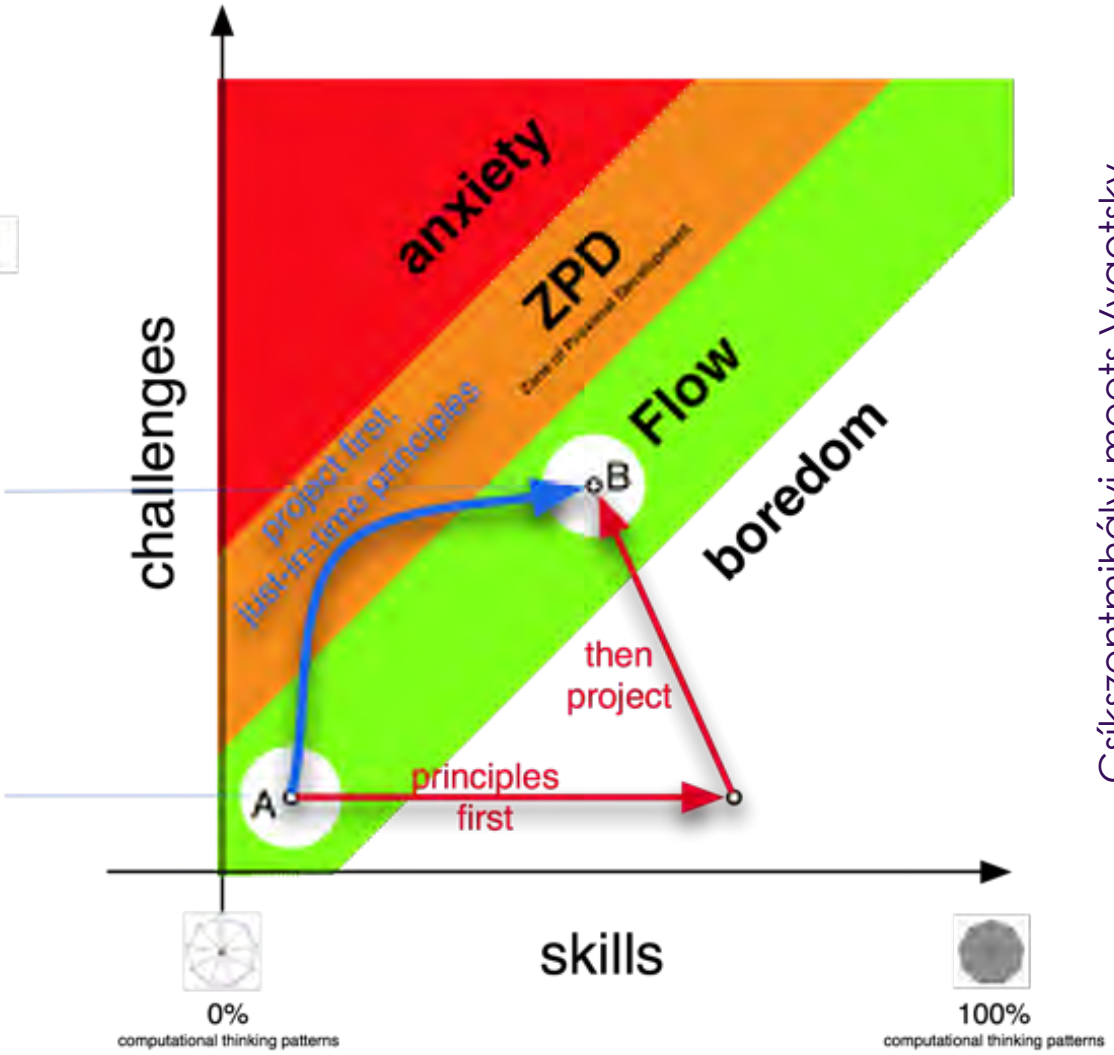
towards a theory of broadening participation



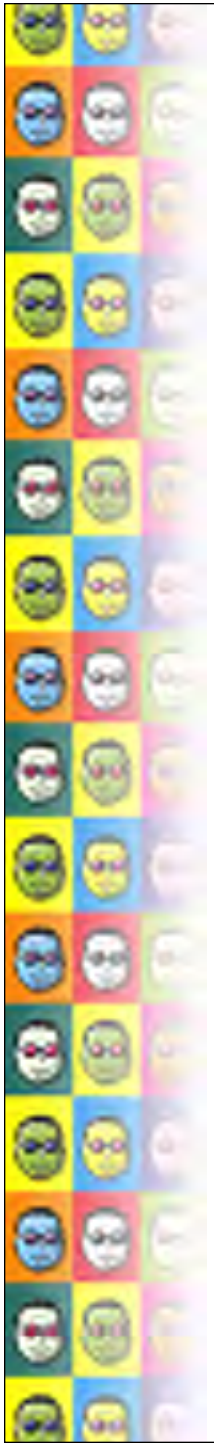
Project First Pedagogy

Zones of Proximal Development



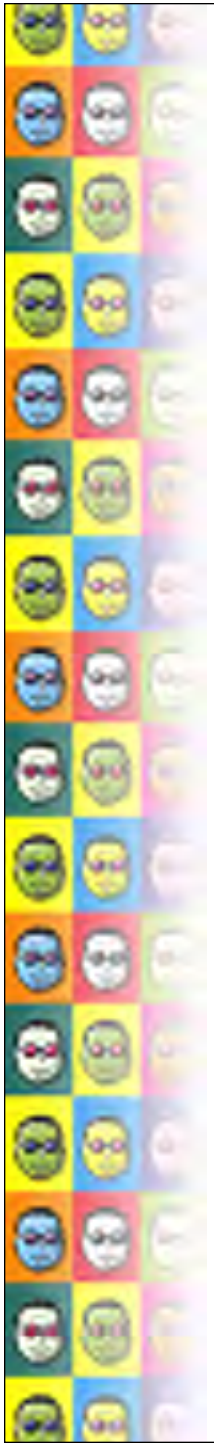


Csíkszentmihályi meets Vygotsky
Zones of Proximal Flow



computing
computational thinking

what can we compute and
what does it mean?



computational thinking

expectation

I want to be able to walk into a classroom with game design and ask a student: "...now that you can make space invaders, can you also make a science simulation?"

– Len Scrogan, Director of Instructional Technology, Boulder Valley School District

the trouble with transfer

- good news: transfer connects well to CSTA, ISTE, NSF operational CT definition
- but:

> This idea--that programming will provide exercise for the highest mental faculties, and that the cognitive development thus assured for programming will generalize or transfer to other content areas in the child's life--is a great hope. Many elegant analyses offer reasons for this hope, although there is an important sense in which the arguments ring like the overzealous prescriptions for studying Latin in Victorian times.
>
> - Roy Pea, in Logo Programming and Problem Solving, 1987

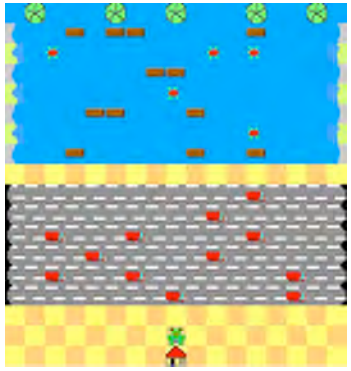
challenges for transfer 2.0

- need to find realistic expectations
- need to find the right level representations, e.g., computational thinking patterns
  - need to make CT patterns computable
  - need to make CT patterns recognizable by people

the “right” level of representation for transfer

game ⟷ science simulation

phenomena

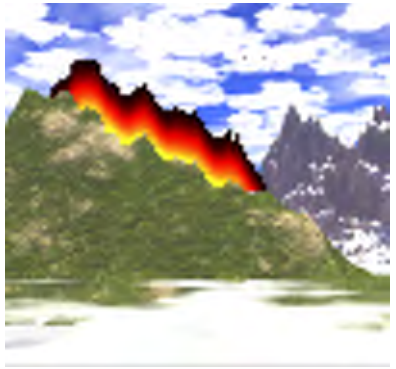


frogger

Michotte: “Perception of Causality”

launch: 

collision, push, pull, diffusion, hill climbing, …



avalanche

program

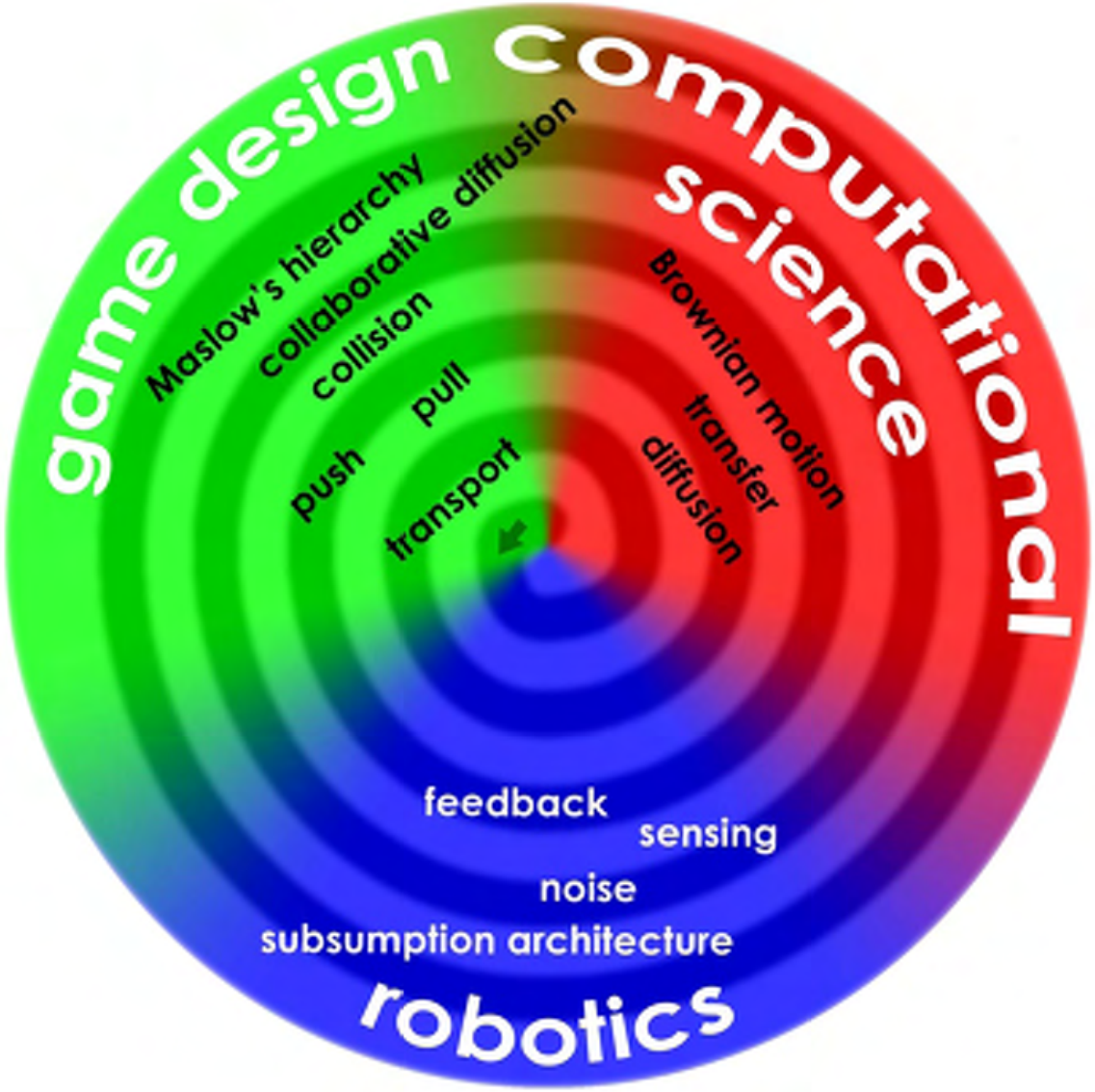
```
def add5(x):
   return x+5

def dotwrite(ast):
   nodename = getNodename()
   label=symbol.sym_name.get(int(ast[0]),ast[0])
   print ' %s [label="%s' % (nodename, label),
   if isinstance(ast[1], str):
      if ast[1].strip():
         print '= %s"];' % ast[1]
      else:
         print '"]'
   else:
      print '"];'
      children = []
      for n, child in enumerate(ast[1:]):
         children.append(dotwrite(child))
      print ' %s -> {' % nodename,
      for name in children:
         print '%s' % name,
```

loop, if, then, else, print, …

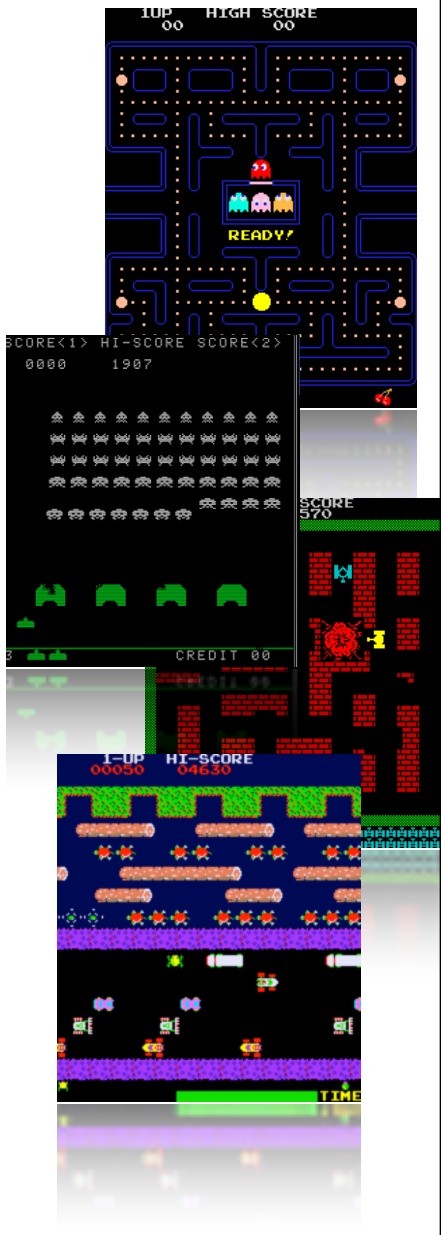
```
import string
import leap2
if __name__=="__main__":
  if len(sys.argv)<2:
    print "Usage:",sys.argv[0],"year year year..."
    sys.exit(1)
  else:
    for i in sys.argv[1:]:
      y=string.atoi(i)
      j=leap2.julian_leap(y)
      g=leap2.gregorian_leap(y)
      if j!=0:
        print i,"is leap in the Julian calendar."
      else:
        print i,"is not leap in the Julian calendar."
      if g!=0:
        print i,"is leap in the Gregorian calendar."
      else:
        print i,"is not leap in the Gregorian calendar."
```

computational thinking inventory



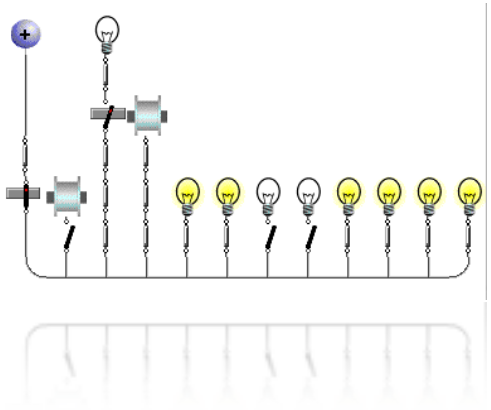
basic computational thinking patterns

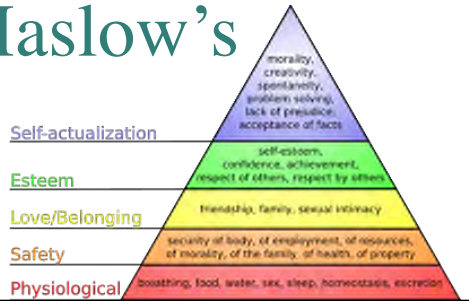
- **Collision**; Frogger: Frog meets Truck
- **Push**; Sokoban: person pushes boxes
- **Transport**: Frogger: logs transport frogs
- **Generate**: Space Invaders: defenders shoot rockets
- **Absorb**: Bridge Builder: tunnel absorbs cars
- **Choreography**: Space Invaders: mother ship makes attack alien ships move left and right and descend
- **Polling / Counting**: Pacman: game over when all the dots are eaten



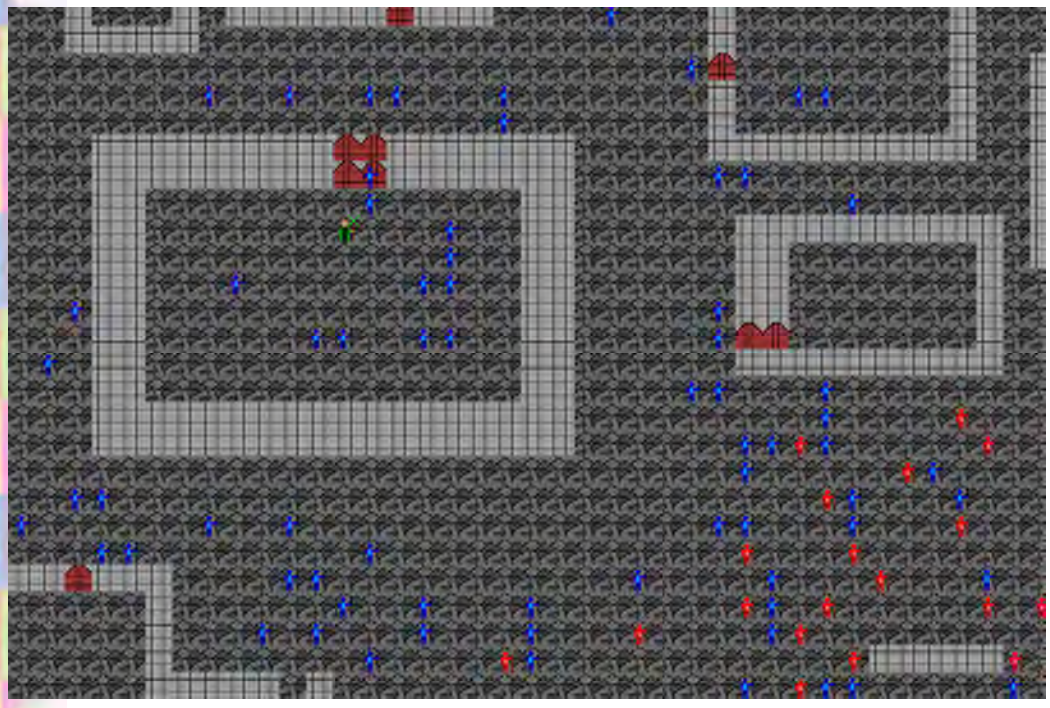
advanced patterns

- **Diffusion**: Electricity, Heat, rumors, toys: spread of information
- **Seeking**: Sims: people finding food
- **Collaborative Diffusion**: Soccer: players collaborate and compete
- **Multiple Needs**: Maslow's hierarchy of needs

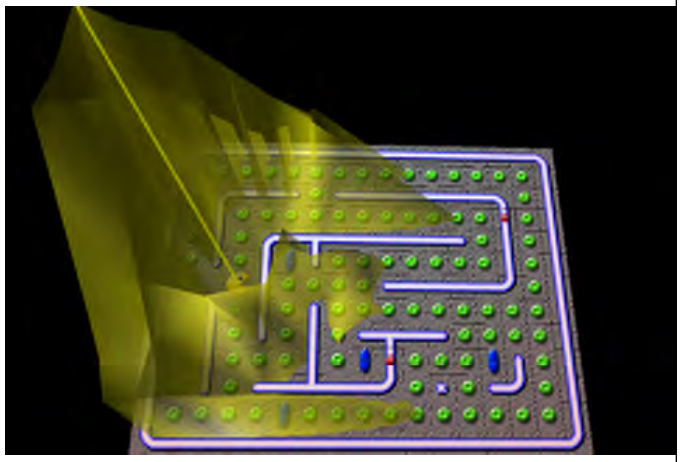




STEM in games



game world

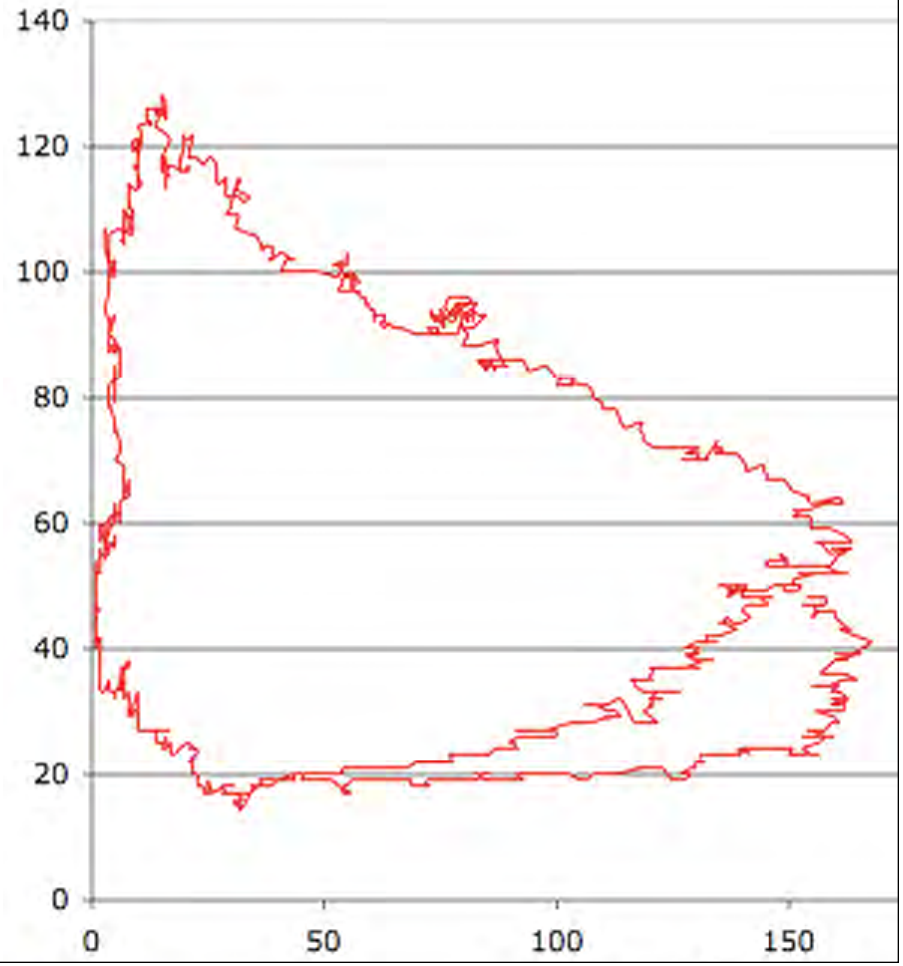


sophisticated visualizations

$$u_{0,t+1} = u_{0,t} + D\sum_{i=1}^{n}(u_{i,t} - u_{0,t})$$

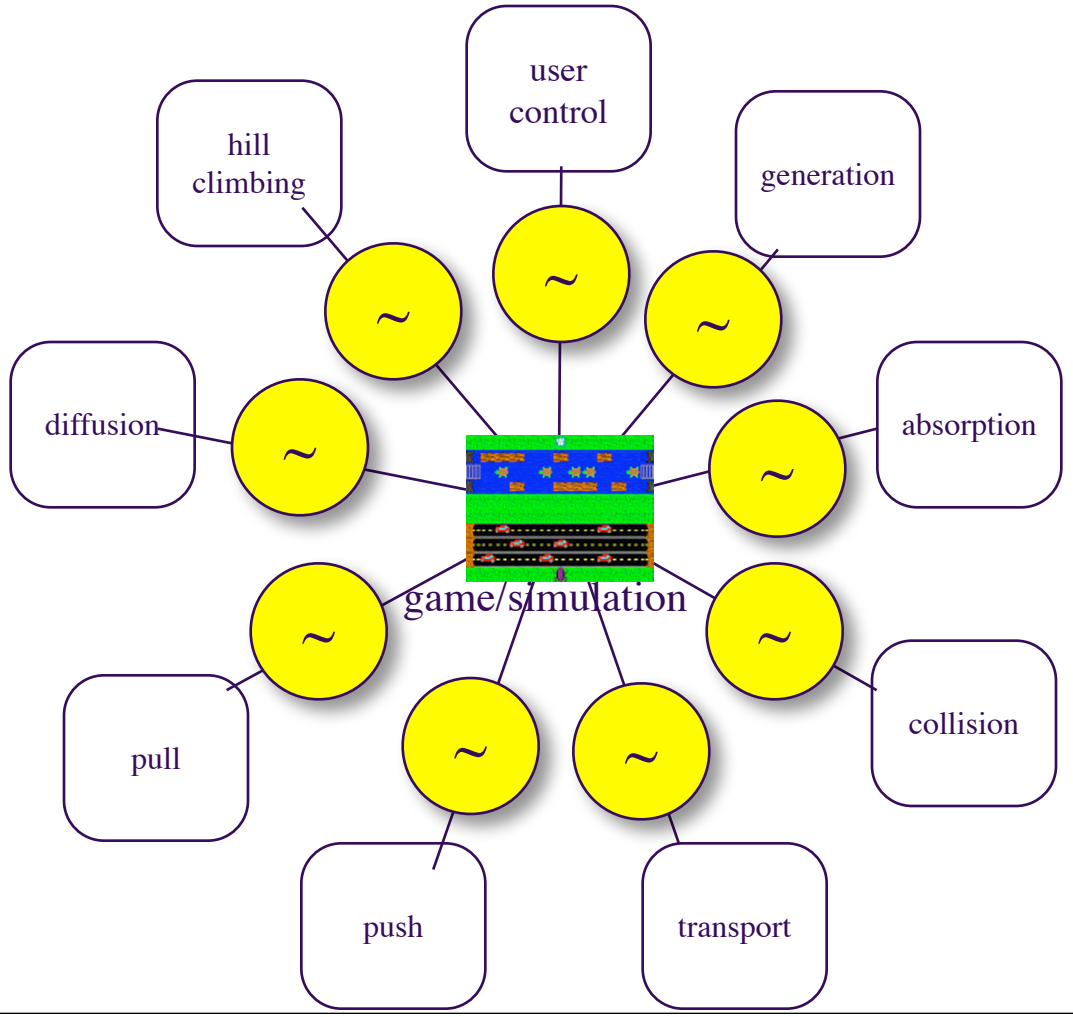
advanced math (diffusion)

transition to computational science models



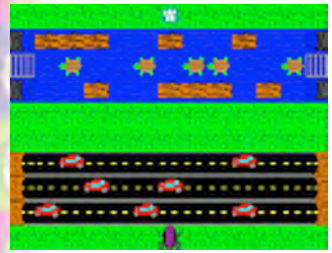

NEWS
8
AUSTIN

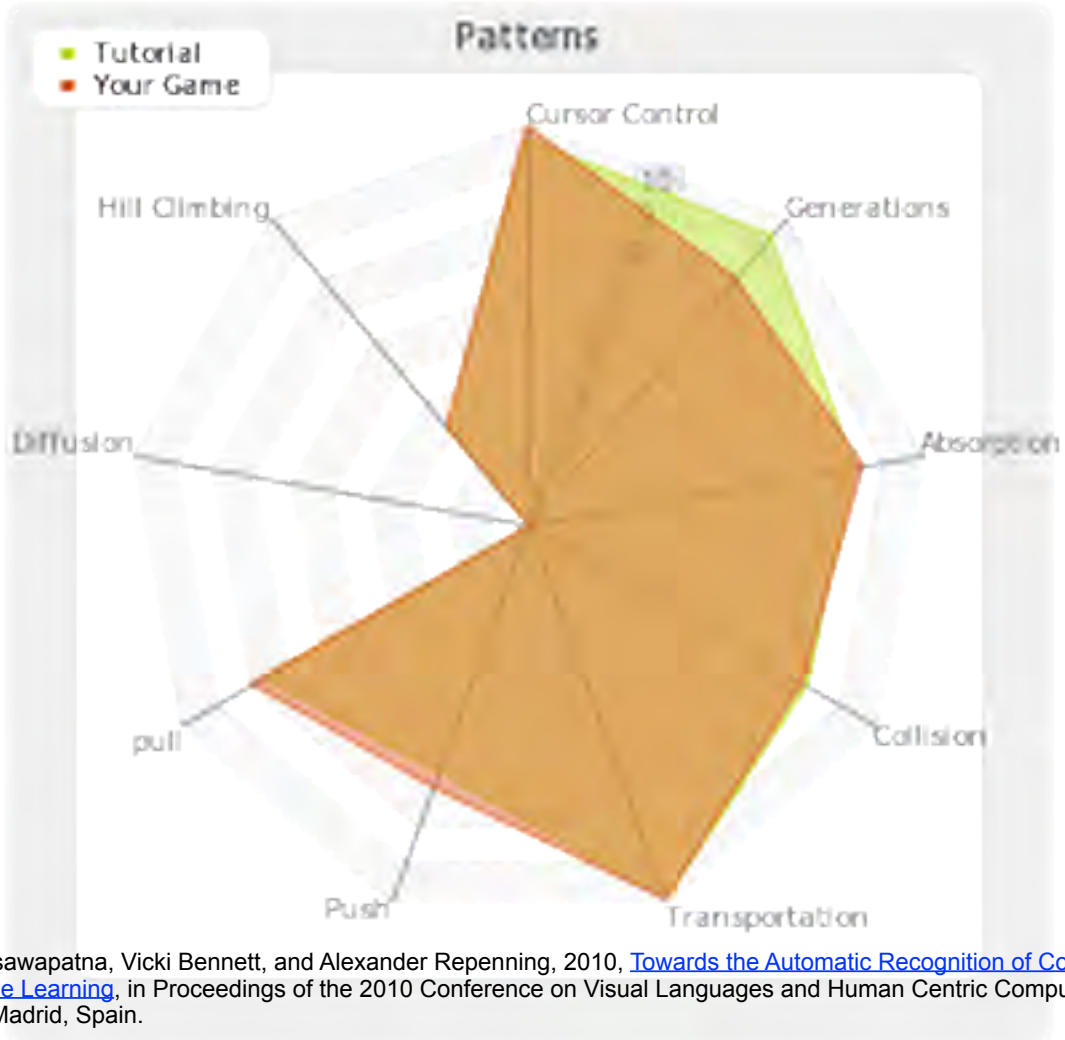
computing computational thinking



Latent
Semantic
Analysis
inspired
similarity

Computational Thinking Pattern Analysis





Kyu Han Koh, Ashok Basawapatna, Vicki Bennett, and Alexander Repenning, 2010, Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning, in Proceedings of the 2010 Conference on Visual Languages and Human Centric Computing (VL/HCC 2010), pp. 59-66, IEEE Computer, Madrid, Spain.

Scalable Game Design Arcade

Computational Thinking Patterns

Probot

Program the robot so it can get through the maze to the flag. The commands that you can use are "forward" "turn left" "turn right" and "change to red" (you need to be red to cross the lava.) use the drawing tool to place the commands into the boxes. Once you have placed the commands into the boxes, press spacebar to have the robot start. R resets everything. To win you must finish the game in 12 move or less, remember, commands on the same line all happen at the same time.

Run

Similarity Score to Four Tutorial Games

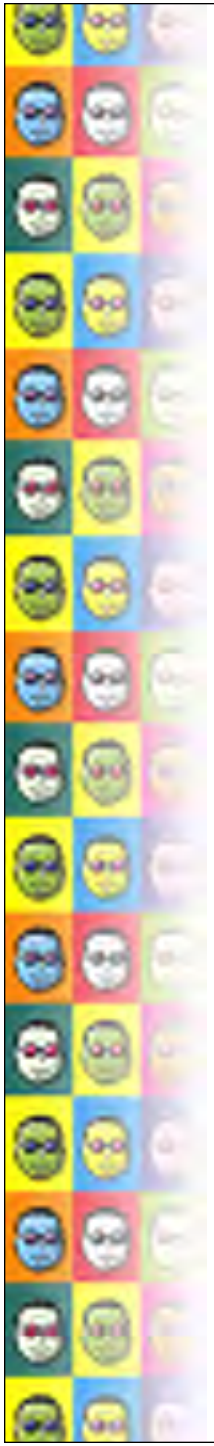
This score shows how much your game structure is similar to the tutorial games. Max value is 1

This game's similarity score to Frogger:0.624
This game's similarity score to Sokoban:0.715
This game's similarity score to Space Invaders:0.644
This game's similarity score to Sims:0.082

Similarity Score Matrix

Below Matrix shows other AgentSheets projects sharing similar programming structure.

This Matrix updates itself every 2 and half hours. It may have random projects right after your submission.

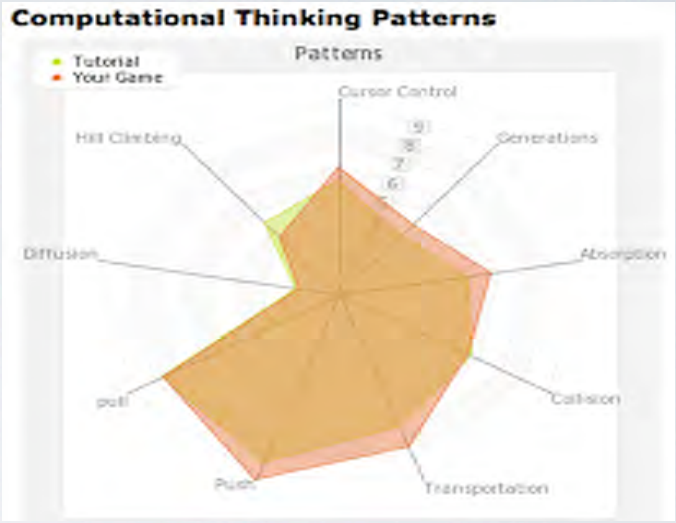


transfer?

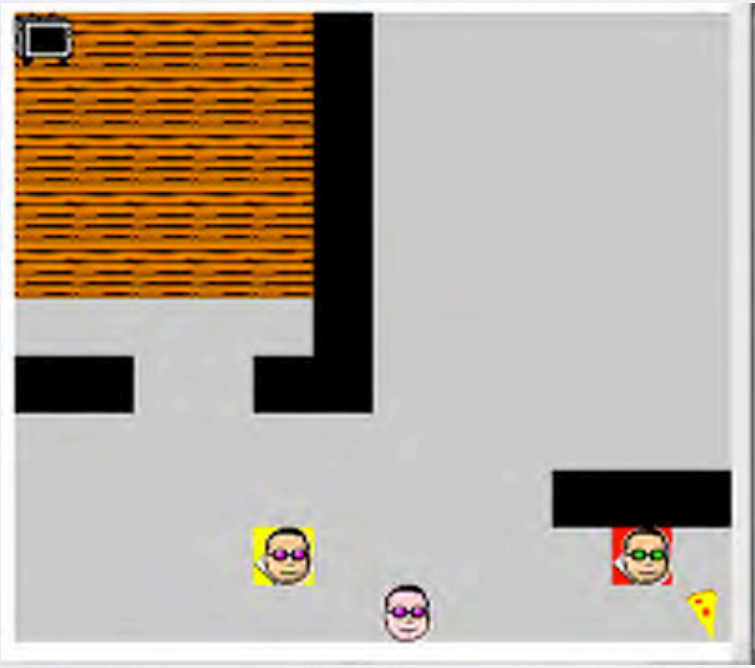
"Now that you can make 'Space Invaders', can you also program a science simulation?"

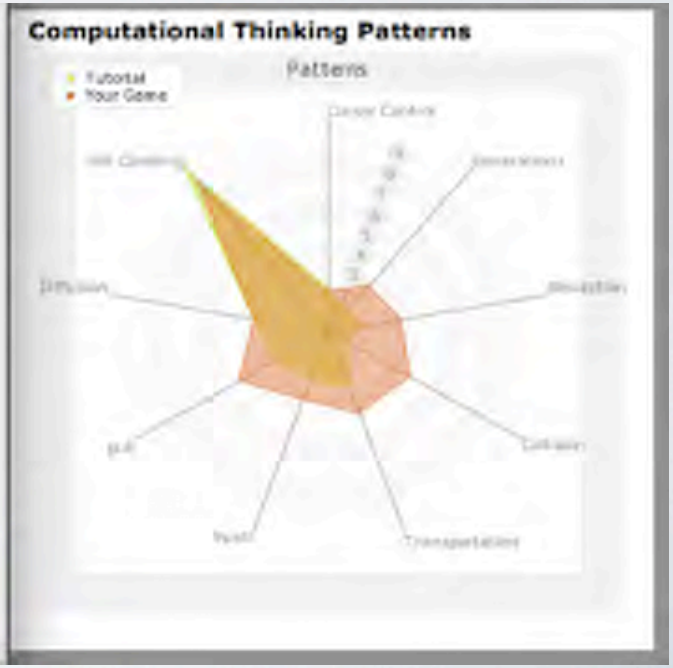
ONE STUDENT MADE DIFFERENT GAMES

Game #1
- Sokoban

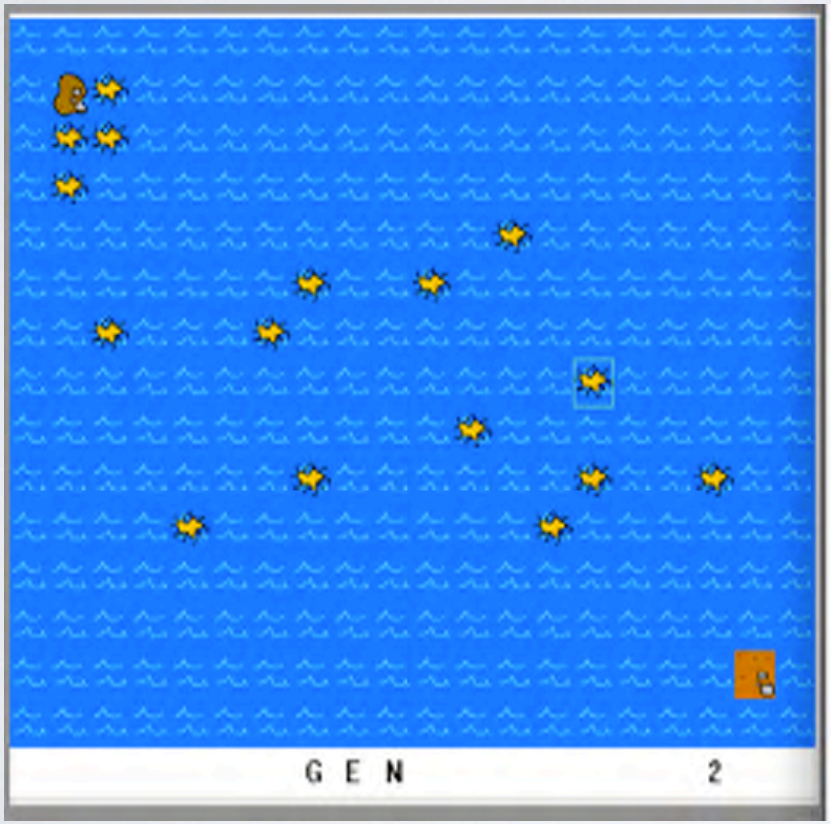


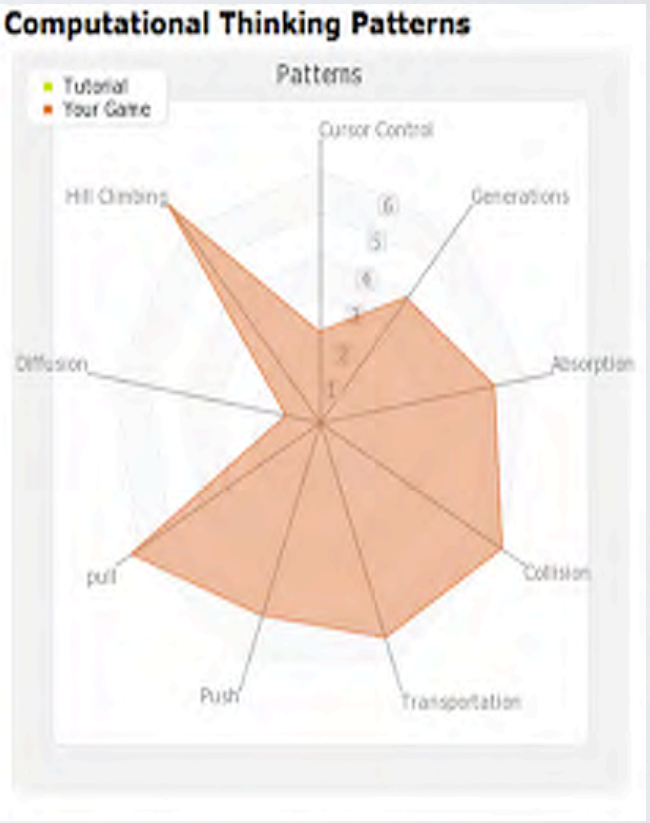
Game #2
- Sims





Science Simulation based on Chaos theory

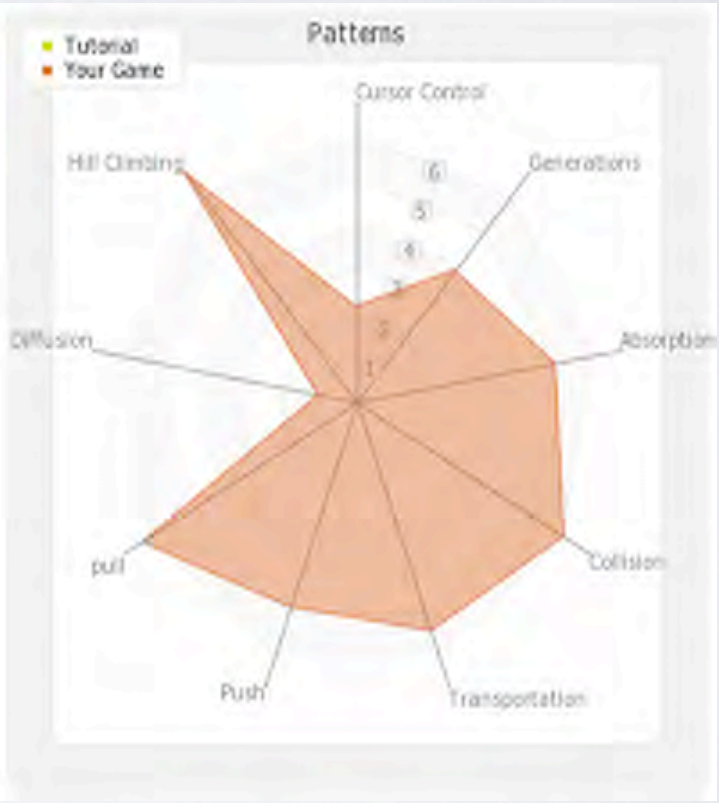


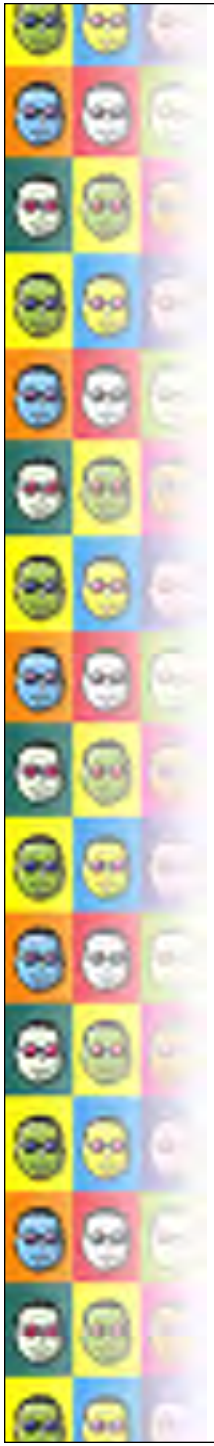


Transfer
From Game Design to Science Simulation Design

Game #1 and #2

Science simulation

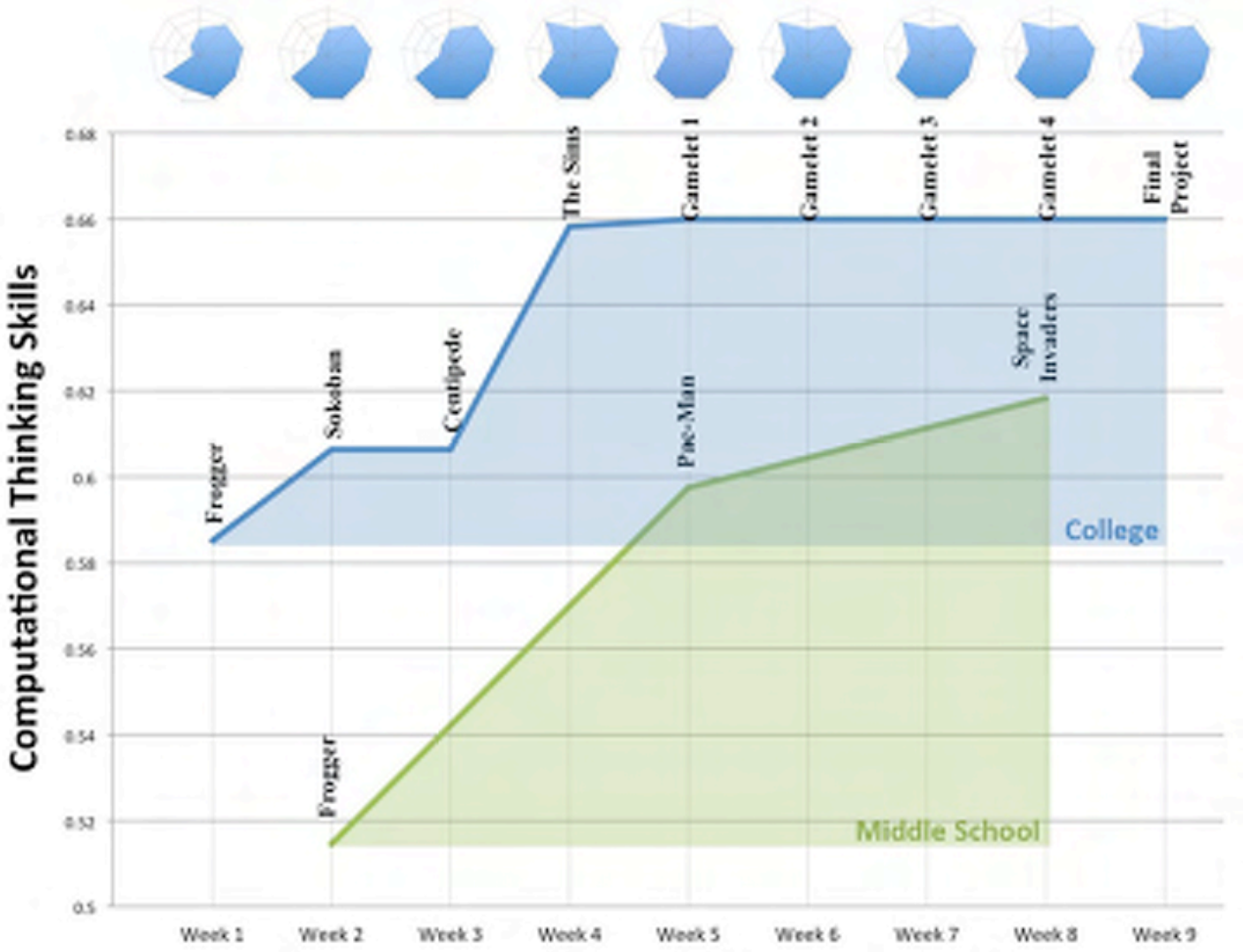




learning trajectories, divergence, ownership and creativity

what else can we compute?

middle school vs. college

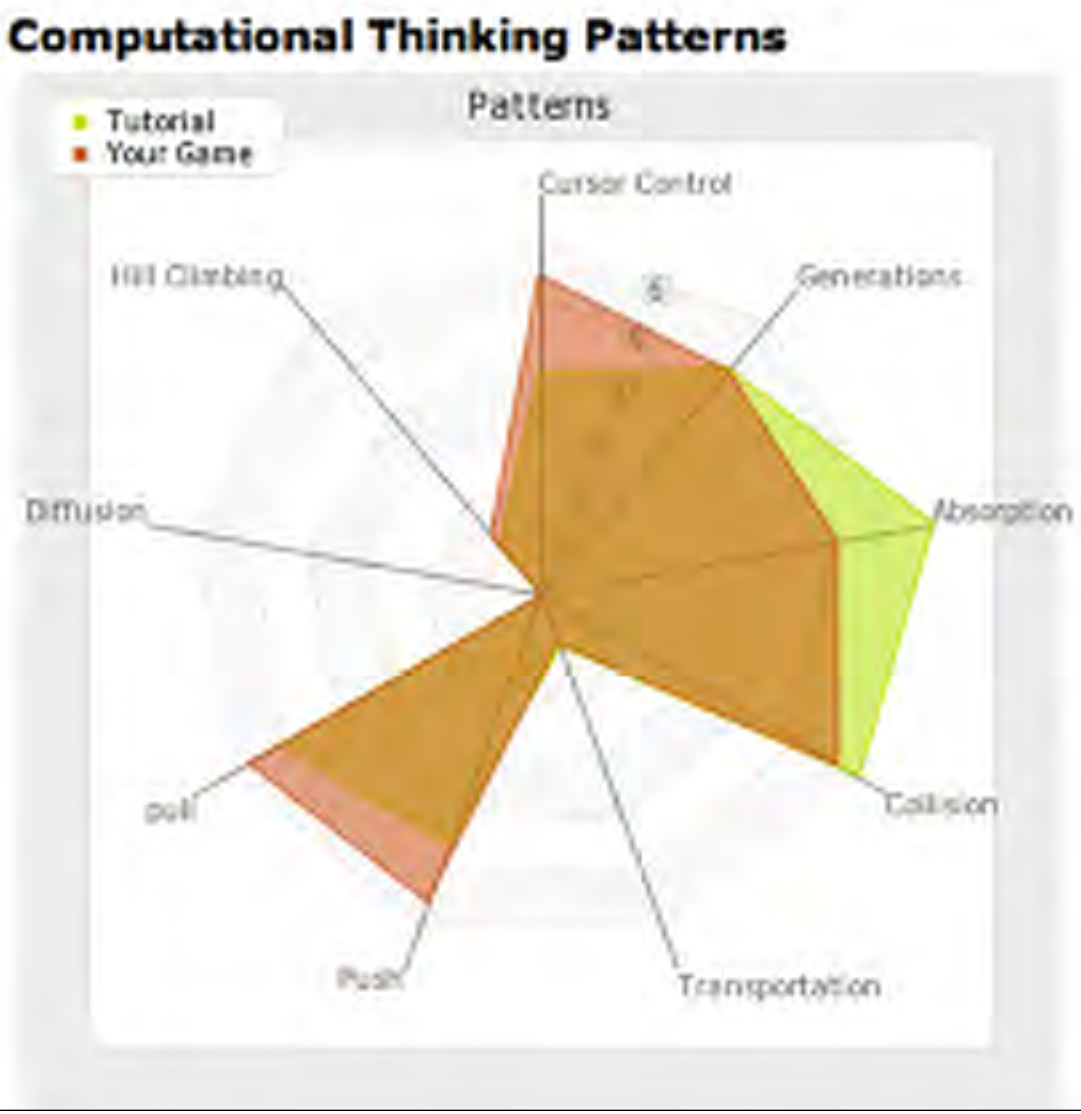


divergence

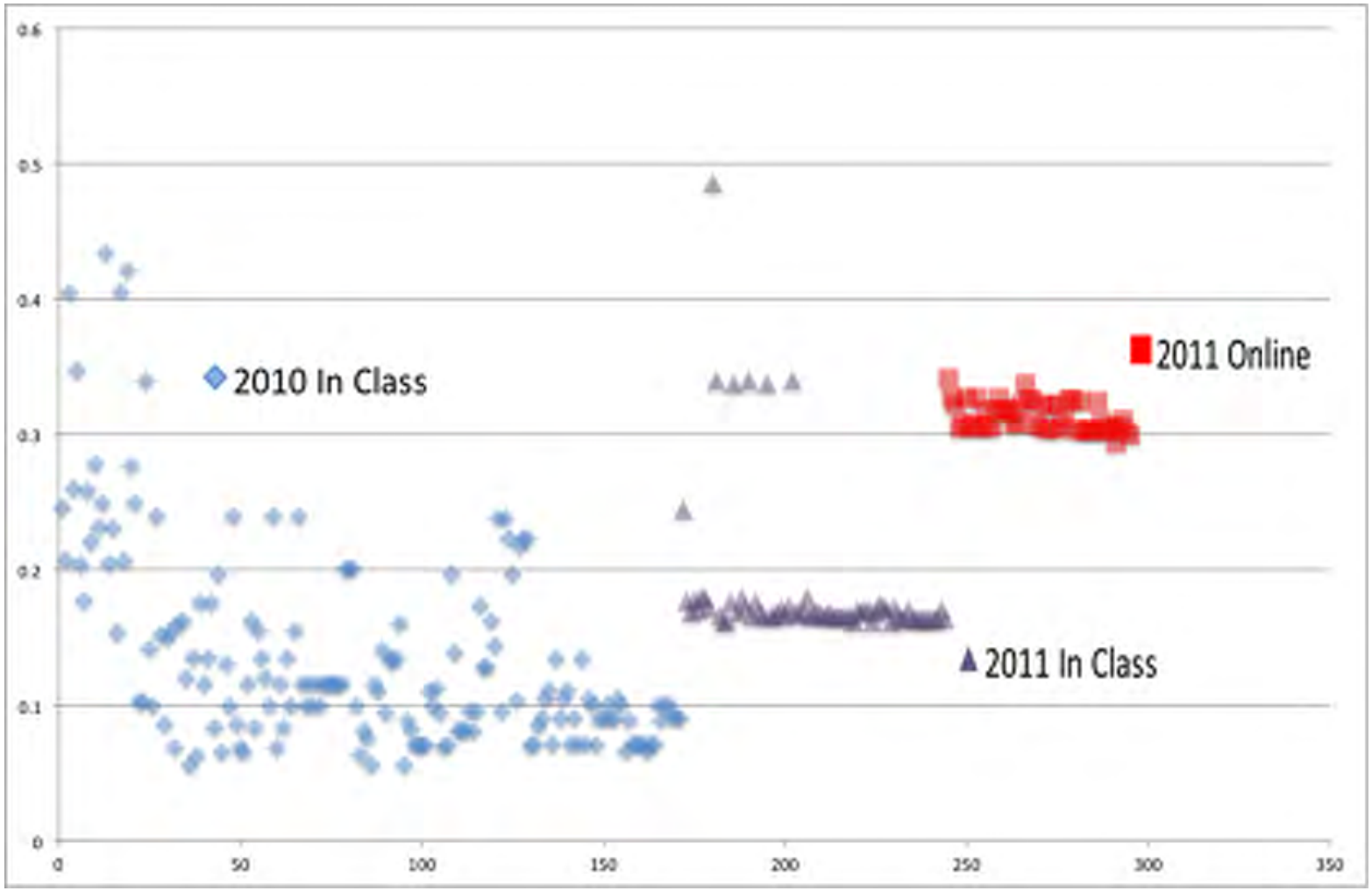
difference between ***Your*** game and ***Tutorial*** game

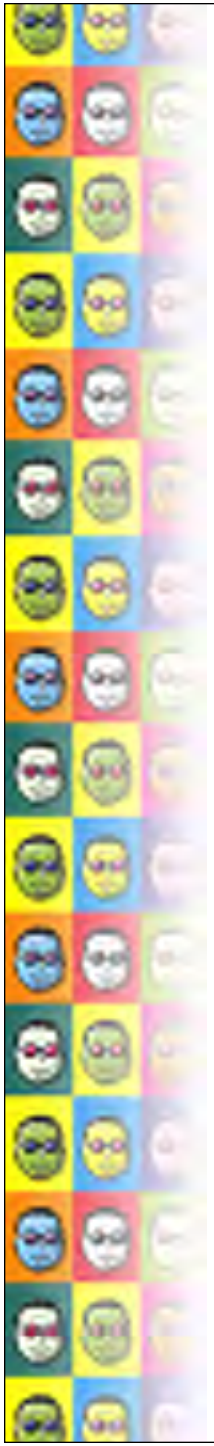
Divergence Score (u, v) =

$$\frac{\sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}}{\sqrt{n}}$$



divergence in Sandy's classes





questions

- divergence ⇒ ownership?
- divergence ⇒ creativity?
- ownership ⇒ creativity?
- what is social divergence/ownership/ creativity?

conclusions

- we can compute computational thinking
- there are early indicators for transfer between game design and science simulation design. But need to be careful:
  - do not confuse correlation with causation
  - investigate role of teacher to scaffold concepts to be transferred.

thank YOU!

SCALABLE GAME DESIGN

http://scalablegamedesign.cs.colorado.edu

Google