SCALABLE GAME DESIGN

Prof. Alexander Repenning
Prof. David Webb
Dr. Jeffrey Kidder
Dr. Andri Ioannidou
Vicki Bennet
project

- **goal**: get computer science into public schools as part of required courses

- **approach**: employ game design to motivate & educate students to learn about computer science through game design starting in middle school

- **research**:
  - equity: gender, underrepresented (Native Americans)
  - transfer to STEM: “now that you can make Space Invaders, can you make a science simulation?”
  - computational thinking (an NSF priority)

- **scope**:
  - years 1-3: strategic exploration of Colorado spectrum
  - potentially: years 4-8: scale up to US.
“programming, oh no... I know what is going to happen. The teacher writes a program onto the blackboard, we type it into the computer and it never works” – student
tool: AgentSheets

AgentSheets ‘93
- KidSim
- Cocoa
- StageCast

Syntax
graphical rewrite rules

AgentSheets ‘95
- Squeak/eToys
- Alice

Syntax
drag & drop

main concern:
help users to think about the meaning of programs → “computational thinking”

Semantics
Conversational Programming

AgentSheets 2010
sites

NSF ITEST project exploring K-12 computer science education in 4 very different kinds of communities
Tech Hub
University Town
Boulder, Colorado
Remote Rural
poorest district in CO
open only 4 days a week
Native American
Southern Ute Tribe
Ignacio, Colorado
Oglala, SD to Ignacio, CO
715 mi – about 12 hours 40 mins drive time

Native American
Oglala Lakota Tribe
Pine Ridge Indian Reservation in South Dakota
scalable lesson plan

- middle school
  - 6th grade: required: first game, one week = 5 x 45min
  - 7th grade: elective: more games, STEM, 4 weeks

- high school
  - transition to Java

- Community College
  - introduction to programming/logic, course

- University
  - ugrad/grad, CS, game design, 1 game/week
  - ugrad, computational science

see our poster in next break
computational thinking tool checklist
computational thinking tool checklist

for systemic impact a computational thinking tool used in K-12 must fulfill all these conditions:

1. **has low threshold**: can make a game in first 2 hours
2. **has high ceiling**: can make a real game that is playable and exhibits sophisticated behavior, e.g., complex AI
3. **scaffolds Flow**: provide educational game design stepping stones with managed skills & challenges
4. **is multipurpose**: works for game design and computational science applications
5. **supports equity**: accessible and motivational across gender and ethnicity boundaries
6. **systemic & sustainable**: can be used by all teachers to teach all students. Support: teacher training, standards alignment
#1 has low threshold
can a middle school teacher with no background in programming/modding/playing games be trained to teach kids to make a complete game in about 3 hours?
simple problem ➔ simple solution?

Frogger:
user can move frog (up, down, left, right) with cursor keys
computational thinking ≠ coding

authoring at *cognitive* level:
most or all components in the solution domain map back to problem domain

- simple ideas ⇒ simple formalization
- intrinsic complexity

authoring at traditional *programming* level: many components in solution domain have no meaning in problem domain.

- ideas ⇒ complex code
- accidental complexity
show me the game!

- A game is more than colorful objects moving around on the screen making noises and speech bubbles.
- Many tools claim to enable students to build games. If tools fall short in building even simple games that their creators and their friends like to play then excitement quickly fades.
- If the tool cannot handle (hard ... impossible) fundamental computational thinking patterns such as transportation (e.g., Alice) then you cannot build even simple 1980 arcade style game such as Frogger.
#2 has high ceiling
“excuse me, I need better AI”

excited students want to move on. The tools need to be able to create advanced games with sophisticated AI

- can you find the shortest path in a maze?
- can you have collaborative agents?
- can you model human motivation (e.g., Maslow hierarchy of needs)?
STEM in games

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

sophisticated visualizations

advanced math (diffusion)
transition to traditional programming

```java
public class frog extends ActiveAgent {

    public synchronized void tasks() {
        if (key(126)) {
            move((byte)1);
            return;
        }
        if (key(123)) {
            move((byte)3);
            return;
        }
        if (key(124)) {
            move((byte)5);
            return;
        }
        if (key(125)) {
            move((byte)7);
            return;
        } else {
            return;
        }
    }
}```
super computing

most scientific explorations require performance

- *fast*: need to be able to run ten thousands of objects
- *parallel*: need to use multiple cores

NASA simulation of eColi, with 2000 agents, using AgentSheets
#3 scaffolds
Flow
Anxiety

Optimal Flow

Boredom

design challenges

design skills

flow
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
#4 is multipurpose
fundamental question of transfer

- school district leaders – barely– tolerate game design but want to see transfer of computational thinking skills to STEM (Science Technology Engineering Math)
- computational thinking definition: expectation
  “now that you can make Space Invaders can you make a scientific simulation?”
- bold claim: if there is not transfer, there is no computational thinking
computational science

- if your math operators are limited to elementary school math you will not get far in STEM
- if you have no plotting, no way to export data and no visualization tools you have no way to analyze computational science models

\[ \sin(\text{time}) \times 0.1 \times \text{value[left]} + 0.9 \times \text{value} \]
computational science model
computational thinking
inventory

middle school

graduate school
#5 supports equity
“I used to only have 2 or 3 girls in my elective classes, now half of the class is girls.”

– Greg Peters, Applied Tech, Computer, Monarch K-8, with participation of 80-100 students per trimester.
the usual suspects

Especially at the high school level existing “cool”
game design courses only attract the usual suspects

- Boulder Valley School District: Progressive schools
  around university town
  - 4 semesters of courses
  - 1 girl

- Jeffco, 84,000 students (CO largest district)
  - 5 semesters
  - 0 girls

- Scalable Game Design (Colorado and South Dakota)
  - 1 semester
  - 52.5% girls (out of over 1300 student participating)
#6 is systemic & sustainable
elective -> required

- computer science education should not be like a field trip to the zoo!
- research needs to move beyond self selected teachers teaching self selected students:
  - crossing the chasm: grassroots individual school -> entire school district
  - ALL teachers
  - ALL students
- examples
  - entire school district: Boulder Valley School District: 12 middle schools
  - entire country: starting to introduce to every school in Greece
align with standards

- ISTE NETS <-> Scalable Game Design
- ACM Model Curriculum for K–12 Computer Science
We are at the tipping point of making systemic impact. For systemic impact a computational thinking tool used in K-12 must fulfill *all* these conditions:

1. has low threshold
2. has high ceiling
3. scaffolds Flow
4. is multipurpose
5. supports equity
6. systemic & sustainable
results

and conclusions
results

- **Real need**: planned for 1200 students in 3 years, already instructed **over 1300 students** in first semester of implementation.
- **Not the usual suspects**: in many K-12 computer science courses participation of girls in computer science courses is less than 10%. This project’s average is **52.3% girls**.
- Some of the participating middle schools instruct over **900 students per year per school**.
- Students want to participate in more game design courses: **78% of girls/68% of boys**
conclusions

With the right combination of computational thinking tools, scaffolding approaches and teacher support, K-12 is right place to begin CS education.

- public schools: to broaden participation through required courses
- after school programs, summer camps: to deepen participation through self selected programs
try out the game design tutorials:

scalable
game design

visit us at poster right after this session

http://scalablegamedesign.cs.colorado.edu
Computational thinking assessment

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.

Submission Time
10/02/21: 17:36:23
first US-wide Native American Game Design class

- summer of 2008
- course filled up immediately
- more women than men
<table>
<thead>
<tr>
<th></th>
<th>Gender</th>
<th>Language</th>
<th>Ethnicity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Boy</td>
<td>Girl</td>
<td>English</td>
</tr>
<tr>
<td>a) I enjoyed designing games on the computer.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Disagree or Disagree</td>
<td>13*</td>
<td>16%</td>
<td>21</td>
</tr>
<tr>
<td>Strongly Agree or Agree</td>
<td>68</td>
<td>84%</td>
<td>133</td>
</tr>
<tr>
<td>Totals</td>
<td>81</td>
<td>100%</td>
<td>154</td>
</tr>
<tr>
<td>b) I am happy with my final game that I designed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Disagree or Disagree</td>
<td>27</td>
<td>33%</td>
<td>41</td>
</tr>
<tr>
<td>Strongly Agree or Agree</td>
<td>54</td>
<td>67%</td>
<td>113</td>
</tr>
<tr>
<td>Totals</td>
<td>81</td>
<td>100%</td>
<td>154</td>
</tr>
<tr>
<td>c) I would like to take another game design class.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strongly Disagree or Disagree</td>
<td>26</td>
<td>32%</td>
<td>36</td>
</tr>
<tr>
<td>Strongly Agree or Agree</td>
<td>56</td>
<td>68%</td>
<td>119</td>
</tr>
<tr>
<td>Totals</td>
<td>82</td>
<td>100%</td>
<td>155</td>
</tr>
</tbody>
</table>
“Drama is life without the boring bits”
— Alfred Hitchcock

“Computational Thinking is programming without the boring bits”