What in the HECK is a Computational Thinking Pattern (CTP)?!
NICE, WE MADE A GAME

So what?

“Now that you’ve made Space Invaders, can you program a science simulation”

-Len Scrogan, BVSD

What transferable skills did we actually learn?
Abstractions students learn in programming activities that transfer to other game and simulation creation activities

**Example Collision Patterns**

**Absorb:** Agent disappears when in contact with another agent

**Change:** Agent Changes when in contact with another agent

A way to make the *Computational Thinking Skills* learned in game programming explicit
Examples of CTPs

Absorb

Game  Real World  Simulation
Examples of CTPs

Transport

Game  Real World  Simulation
Computational Thinking Patterns

Common high-level agent-interactions that students implement to create both games and simulations
Why do we care about them?

As students are learning how to create their Frogger game, for example, they are also learning the Computational Thinking Patterns necessary to create simulations.

As students were creating games they were actually learning the skills necessary to create simulations.
Here are some example animated Generic Computational Thinking Patterns
Collision Patterns

- **Change:** An Agent Changes One Agent Into Another Agent
  - Examples: Pac-Man Becoming Invincible After Eating A Power-Pellet, Mario Becoming Big After Eating A Mushroom, Truck Hitting A Frog Changing It Into A Dead Frog, Sick Student Making Another Student Sick etc.

- **Absorb:** One Agent Makes Another Agent Disappear
  - Examples: An Animal Eating Food, Bullets Hitting Bad Guys Making Them Disappear etc.

- **Transport:** One Agent Rides On Another Agent
  - Examples: Frog In Frogger Riding On A Log, Blood Cells Carrying Oxygen, Dolphin Carrying a Person, Person Carrying A Key, Ants Bringing Food Back To The Hill etc.

- **Push:** One Agent Pushes Another Agent
  - Examples: Person Pushing A Box Or A Closed Door etc.
**Movement Patterns**

**Random Movement: An Agent Randomly Moves Around The World**
Examples: Bugs ‘Randomly’ Buzzing Around A Level, People ‘Randomly’ Walking Around A City etc.

**Tracking: One Agent Chases Another Agent**
Examples: Fox Chasing Rabbits, Ghosts Chasing Pac-man, Person Going To Fridge When Hungry etc.

**Keyboard Movement: Use The Keyboard To Control An Agent's Movement**
Examples: Anytime You Want The Player To Control Movement-- Moving Mario Or Moving The Frog In Frogger Around The World etc.

**Directional Movement: Agent Moves In One Direction**
Examples: Trucks And Logs Moving Across Screen In Frogger, Bullets Flying Through A World etc.
Generation: One Agent Creates Another Agent

Examples: Tunnel Creates Trucks Every So Often, Gun Generates Bullets On A Keyboard Hit, Two Agents Mate Creating Another Agent When Next To Each Other etc.
Why these patterns?

Not exhaustive

Humans decipher these agent-interaction patterns naturally
Albert Michotte: The Perception of Causality

PHENEMENOLOGY

When the timing is just right, people perceive causal relationships.
Computational Thinking

Patterns as Abstractions

Think of “Abstractions” as the stuff we ignore in order to represent.

To make this jump we Abstract out the agents, and preserve the interaction.

i.e. these two things are similar if we ignore the agents involved.
“In working with rich abstractions, defining the ‘right’ abstraction is critical. The abstraction process—deciding what details we need to highlight and what details we can ignore—underlies computational thinking”

--Jeanette Wing
By explicitly teaching these abstractions (i.e. note the placards on the walls), the game or simulation design process can be broken down into identifying which CTP’s should we use…

Change: An Agent Changes One Agent Into Another Agent
Examples: Pac-Man Becoming Invincible After Eating A Power-Pellet, Mario Becoming Big After Eating A Mushroom, Truck Hitting A Frog Changing It Into A Dead Frog, Sick Student Making Another Student Sick etc.

Absorb: One Agent Makes Another Agent Disappear
Examples: An Animal Eating Food, Bullets Hitting Bad Guys Making Them Disappear etc.

Transport: One Agent Rides On Another Agent
Examples: Frog In Frogger Riding On A Log, Blood Cells Carrying Oxygen, Dolphin Carrying a Person, Person Carrying A Key, Ants Bringing Food Back To The Hill etc.

Push: One Agent Pushes Another Agent
Examples: Person Pushing A Box Or A Closed Door etc.

Random Movement: An Agent Randomly Moves Around The World
Examples: Bugs Randomly Buzzing Around A Level, People Randomly Walking Around A City etc.

Tracking: One Agent Chases Another Agent
Examples: Fox Chasing Rabbits, Ghosts Chasing Pac-man, Person Going To Fridge When Hungry etc.

Keyboard Movement: Use The Keyboard To Control An Agent’s Movement
Examples: Anytime You Want The Player To Control Movement-- Moving Mario Or Moving The Frog In Frogger Around The World etc.

Directional Movement: Agent Moves In One Direction
Examples: Trucks And Logs Moving Across Screen In Frogger, Bullets Flying Through A World etc.
We can break down games and simulations into constituent Computational Thinking Patterns
What Computational Thinking Patterns (skills) do my students need to master to meet the challenge of making Frogger? Forest Fire? Pacman?
WE CAN ALSO USE COMPUTATIONAL THINKING PATTERNS TO **ASSESS**
CTPA DIAGRAM
Show the proportionality of patterns in a game

CTP’s Gives us a way to **measure** **SKILL**
Computing Computational Thinking

Latent Semantic Analysis inspired similarity

CTPA (m) = \[
\left[ \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}} \right]_1
\]
WHAT IF WE COULD ASSESS STUDENTS IN REAL TIME

1. sets challenge
   - Teacher

2. instructs students
   - Teacher

3. students build games
   - Students

4. receives real-time feedback
   - Assessment Dashboard (iPad)

Computational Thinking Pattern Analysis of all games

- Anxiety
- ZPD
- Flow
So as you go through games and simulations with your class, break them down by CTP’s