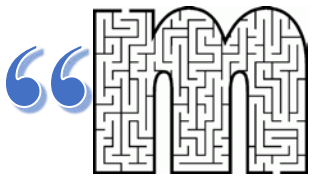
Creating "Maze Craze"



.

Created by: Susan Miller, University of Colorado, School of Education

This curricula has been designed as part of the Scalable Games Design project.
It was created using ideas from and portions of prior work completed by
Fred Gluck.

This material is based upon work supported by the National Science Foundation under Grant No. DRL-1312129 and CNS-1138526. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

SCALABLE GAME DESIGN

Maze Craze

Lesson Objective:

- **To create a game of Maze Craze**
- **To master the basic methods of AgentSheets software**
- **To be able to describe and apply the Computational Thinking Patterns identified below.**

Prerequisite Skills:

- **Basics of computer handling**
- **Identifies hardware components eg. keyboard, mouse, monitor/screen**
- **Identifies cursor**
- **Recognizes the typical features of an applications window title bar, toolbar, menu bar, status bar, scroll bar**
- **Has a knowledge of drop-down menus**
- **Selects menu items from a drop-down menu**
- **Starts an application and creates a document**
- **Names and saves a document in appropriate folder/directory**

Computational Thinking Patterns:

- **Cursor Control**
- **Collision**

Length of Activity:

- **One to Two 30-45 minute lessons, although some students may advance more quickly**

Activity Description:

- **Part 1: Create basic world (worksheet) and the agents**
- **Part 2: Program the traveler and the Chaser**

Challenge activities for students who finish early

Table of Contents

Teacher Instructions: Basic Game

Student Handout 1A: Basic Game – General Instructions

Student Handout 1B: Basic Game – Explicit Instructions

Student Handout 1C: Agent Creation Models

Student Handout 2: Challenge

Vocabulary/Definitions

(Found on page 2 in the student packet)

Action .....the requested behavior of an agent if the conditions are true

Agent .....a character in the game

Array .....a rectangular arrangement of agents

Collision .....the situation when two agents physically collide.

Condition .....the situation that must be 'true' for an action to occur

Depiction.....an image of the agent.

General Teaching Strategies[1]

Basic Philosophy

- The educational goal of these lessons is to learn and apply Computational Thinking Patterns in the context of a familiar game. Emphasis on these Computational Thinking Patterns is essential for student understanding.

- Every effort has been made to create instructions with an eye toward guided discovery. Direct instruction has been used for those aspects where students are learning the code for the first time; however, materials have been provided to ensure that students are understanding the programming concepts, as opposed to simply copying code. Note that special materials have been designed for students who are new to this program.

- Student materials are available for each portion of the game design. These materials are intended to be used in addition to teacher materials, which provide prompts and discussion points. Students may become frustrated with too little teacher support. Students may lose out on conceptual understanding with too much teacher support.

Guided Discovery Process

- **Model the process** rather than just giving students the answer. Building the game on your own, before trying it with your students will enable you to see possible struggling points.

- Have students work through problems on their own. Ask directing questions or give helpful suggestions, but **provide only minimal assistance** and only when needed to overcome obstacles.

- Don't fear **group work**! It is common for computer programmers to talk through problems with one another, and to use code snippets found from other programs, and other programmers. Talking through coding problems enables students to think more

[1] This information is supported by research found in the following documents:

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010, June). Using scalable game design to teach computer science from middle school to graduate school. In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (pp. 224-228). ACM.

National Research Council. (2011). *Learning science through computer games and simulations.* (M. Hilton & M. Honey, Eds.). Washington, DC: The National Academies Press.

National Research Council. (2014). *STEM Integration in K-12 Education:: Status, Prospects, and an Agenda for Research*. (M. Honey, G. Pearson, & H. Schweingruber, Eds.). Washington, DC: The National Academies Press.

Repenning, A., & Ioannidou, A. (2008, March). Broadening participation through scalable game design. In *ACM SIGCSE Bulletin* (Vol. 40, No. 1, pp. 305-309). ACM.

critically about Computational Thinking Patterns, as well as the steps needed to solve a problem. Additionally, seeing how others solved an issue with code helps students realize that problems often have multiple solution strategies, and some that might be more effective than others

- Recognize that programming is largely a process of **trial and error**, particularly when first learning. It is helpful to encourage this mindset with your students.

Building Blocks

- Each project is designed to build on the prior one. Very little student support is provided where expertise has already been created. Conversely, material that is new has more support.

- Be sure to talk through the building blocks (especially for PacMan in the area of diffusion and hill climbing) as these Computational Thinking Patterns will appear often in future games and simulations.

- Remember that conceptual understanding takes time, and it may be necessary to explain these concepts multiple times, using different examples, so that all students can be successful.

Support Learning

- Research shows that game design is associated with engaged students, and engaged students show higher levels on conceptual understanding. Allowing students to personalize their games aids in this engagement and motivation.

- Coding may be difficult for some students, and all students are likely to be frustrated at times when the code does not produce the expected results. **Praise students** for sticking with the troubleshooting process and encourage them to share what they learned with others.

- Be sure to communicate that **the process is more important than the answer**, and that coding of a project often takes time. Do not place pressure on your students to 'hurry up' and resort to giving them the code. The process of figuring it out on his/her own will result in much stronger conceptual understanding.

Differentiated Instruction

Note that there are many vocabulary words in this lesson that may be new for your students. Take time to define those words. Using the words in context often will reinforce their meaning for the students.

- **Students who need a challenge:** Some students with more fluency in programming may finish this very quickly – be prepared for them to move on earlier than other students by having student materials ready in advance.

- **Students who need more assistance:** Other students (especially those with no Maze Craze experience) may struggle a bit more. There are two options for differentiated instruction. Consider the needs of the student and the class as you decide which will work best.

    - Option 1: Pair a struggling student with an experienced student
    - Option 2: Provide student with the ALTERNATIVE student packet. Note that these instructions are more explicit and do not support as much independent thinking. As a result, they should only be used when absolutely needed.
    - Vocabulary for ELL Students: : Collision, Agent, Depiction, Condition, Action
    - Time management issues: While students can be more engaged when they design their own agents, some students can spend too much time on this design or find it frustrating. Handout 1C provides block images of each agent as portrayed in this lesson.

Teacher Instructions:

Part 1 – Create Worksheet and Agents

Teacher note: Maze Craze is an easier version of Journey, and can be used either as a first game (before Frogger) or as a preliminary step to Journey or Contagion. The STANDARD student packet presumes prior AgentSheets experience. The ALTERNATIVE student packet is for students with no prior AgentSheets experience.

Introduce this project to the students by describing the game, Maze Craze.

> *The traveler will walk around on the ground surrounded by walls. The object of the game is to move next to the goal without moving next to one of the Chaser agents. If you reach the goal, the game ends happily. If you move next to a Chaser agent or vice versa before reaching the goal, the game ends unhappily.*

- Ask students to explain to the person next to them how the game works, and the rules of the game.

Ask your students…
• What agents are part of this game? • What does the traveler have to do? • What do the Chasers do? • How do you win the game? • How do you lose the game?

Explain that these are all design features that must be considered when planning a game. Now, tell them that they will be designing their own Maze Craze game.

As a class, briefly create a description of the Maze Craze game similar to the one below.

- identify game objects, called agents, by locating nouns in the game description

  > *The traveler will walk around on the ground surrounded by walls. The object of the game is to move next to the goal without moving next to one of the Chaser agents. If you reach the goal, the game ends happily. If you move next to an Chaser agent or vice versa before reaching the goal, the game ends unhappily.*

- categorize agents into user controlled agents, agents that move or do other things by themselves (sometimes also called artificial intelligence agents) and completely passive agents acting as props such as the ground.

User controlled agents:

- *Traveler*

Artificial Intelligence Agents:

- *Goal, floor, walls, Chaser*

- identify agent interaction by locating verbs in the game description

  The traveler will walk around on the ground surrounded by walls. The object of the game is to move next to the goal without moving next to one of the Chaser agents. If you reach the goal, the game ends happily. If you move next to an Chaser agent or vice versa before reaching the goal, the game ends unhappily.

NOTE: When a chaser is NEXT TO a traveler, that is a form of COLLISION. Make sure your students understand this is a computational thinking pattern. This is difficult for some students as they may be literal about the word collision. This is an important conceptual point to discuss.

Provide Handouts to students.

- **STUDENT HANDOUT 1A Provides general instructions for creating the agents and the worksheet, and programming behaviors for the agents**
  - ***(Found on page 3 in the STANDARD student packet)***
- **STUDENT HANDOUT 1B Provides step by step instructions for creating the agents and the worksheet and programming behaviors for the agents**
  - ***(Found on page 3 in the ALTERNATIVE student packet)***
- **STUDENT HANDOUT 1C shows possible designs for each agent, but encourage students to create their own designs if time allows.**
  - ***(Found on page 6 in the STANDARD and page 11 of the ALTERNATIVE student packet)***

Maze Craze

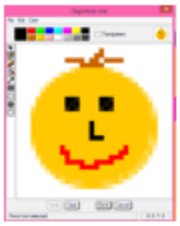
Student Handout 1A:

Basic Game

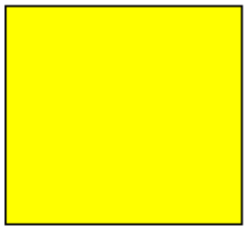
Initial Story: *The traveler will walk around on the ground surrounded by walls. The object of the game is to move next to the goal without moving next to one of the Chaser agents. If you reach the goal, the game ends happily. If you move next to an Chaser agent or vice versa before reaching the goal, the game ends unhappily.*

Create these Agents:

Me



Floor

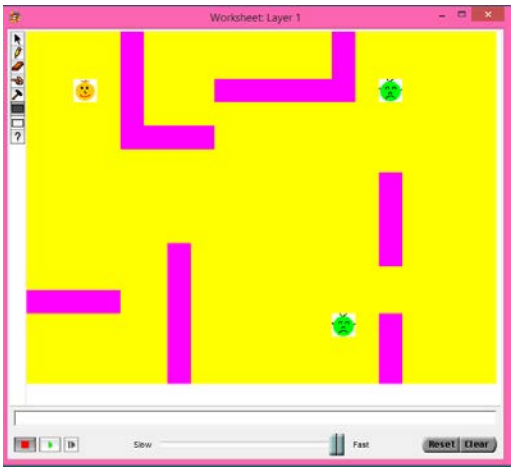


Wall

Goal

Attacker

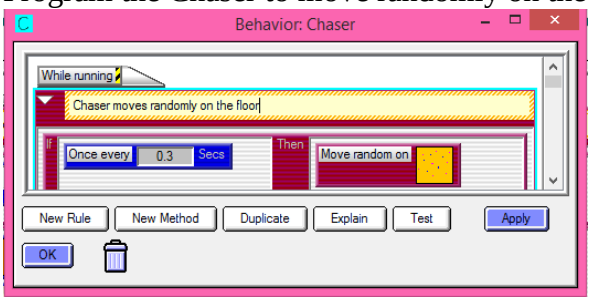
Create this initial Worksheet:



Maze Craze (Continued)

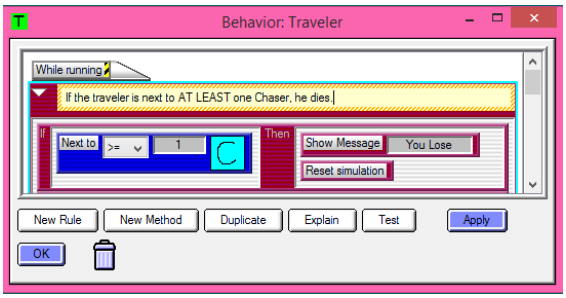
Create the following BEHAVIORS for your agent:

Step 1: Chaser:
Program the Chaser to move randomly on the floor.



Step 2: Traveler:
Set up your agent to move with the arrows (cursor control).
Create game ending conditions (collision).

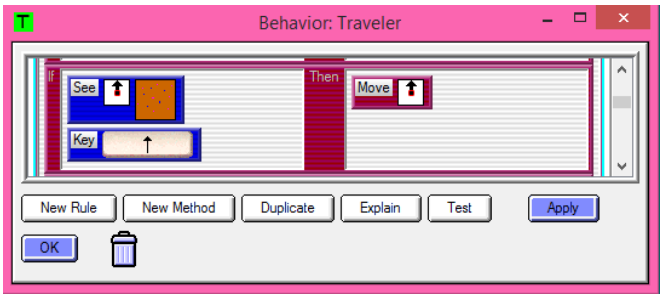
The box below shows how to end the game if your traveler approaches the chaser. Create a similar rule if your traveler approaches the goal.



Be SURE to reset the game when it ends.

Step 3: Walls
Add walls to your worksheet. Then, prevent your Traveler from walking through the walls.

Work with the person next to you to figure out how to prevent the Traveler from walking into a wall. Here is one way to think about it… challenge yourselves to find a different way.



Step 4: Don't allow the traveler to cheat!

Problem: Traveler can cheat by moving off the game ground. Talk with the person next to you about where and when this can happen on your worksheet.

Add rules that make a sound for attempted movement off the ground. Note the importance of rule order for the new rules. Here is an example to prevent the Traveler from moving right off the worksheet. What other direction limit will you need?

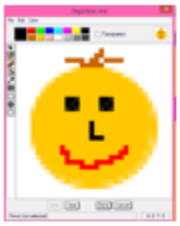
Student Handout 1B:

Basic Game

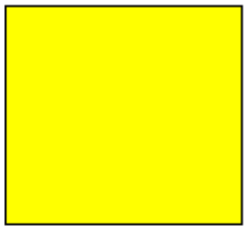
Initial Story: *The traveler will walk around on the ground surrounded by walls. The object of the game is to move next to the goal without moving next to one of the Chaser agents. If you reach the goal, the game ends happily. If you move next to an Chaser agent or vice versa before reaching the goal, the game ends unhappily.*

Create these Agents:

Me



Floor

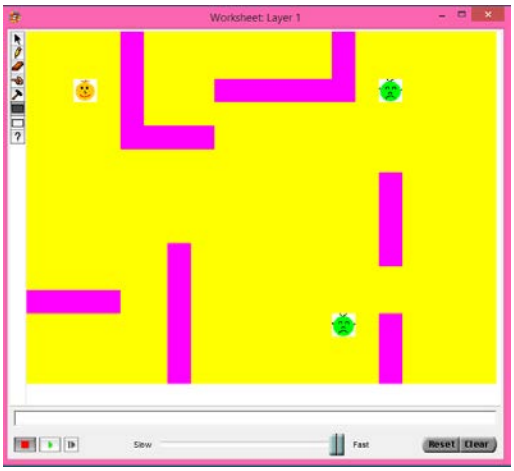


Wall

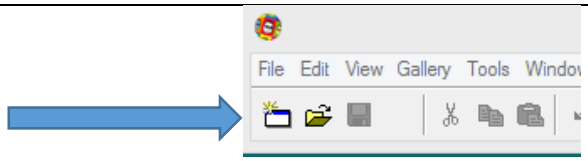
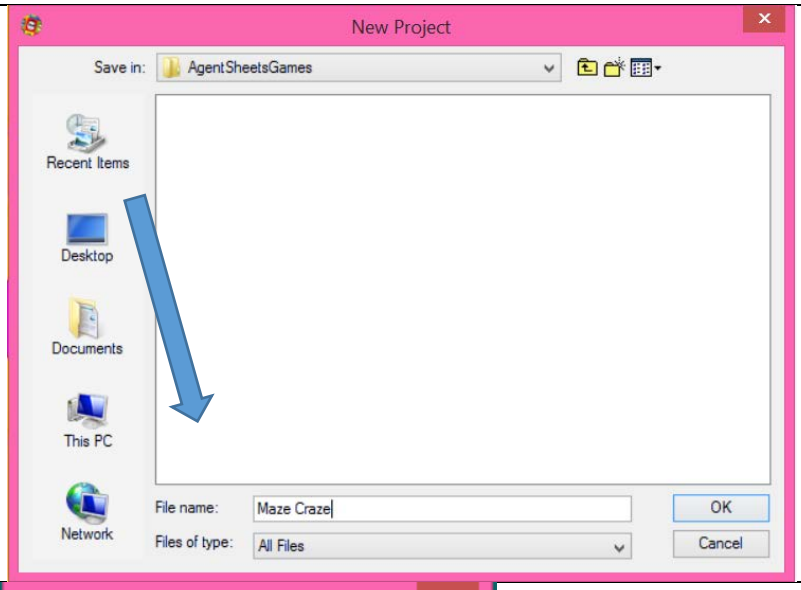
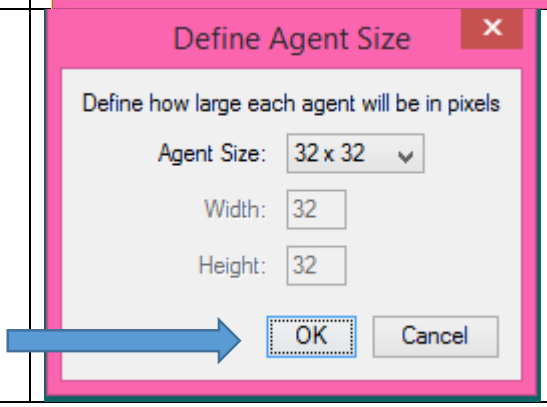
Goal

Attacker

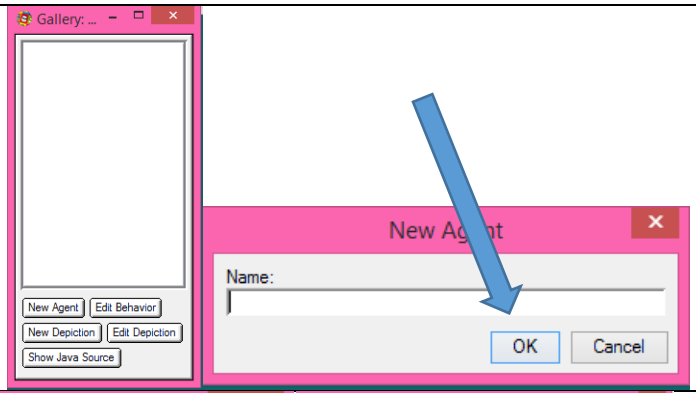
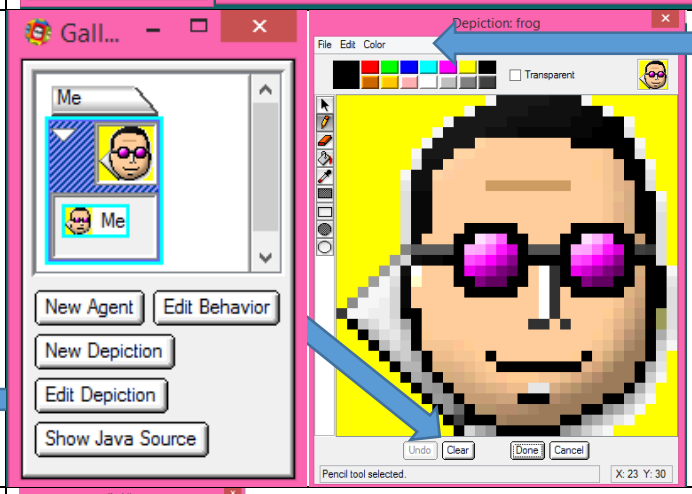
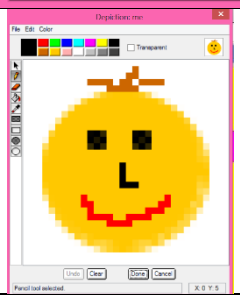
Create this initial Worksheet:



Student Handout 1B: Create a game

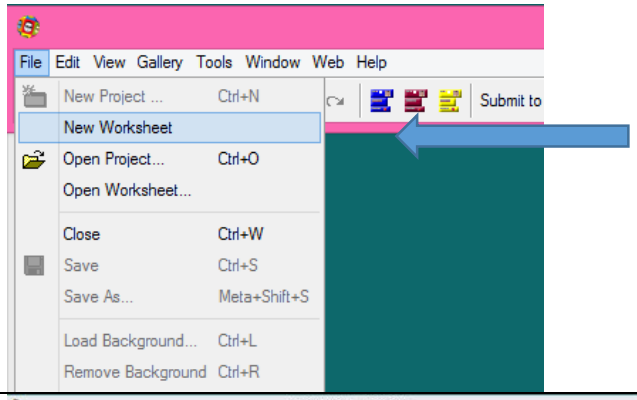
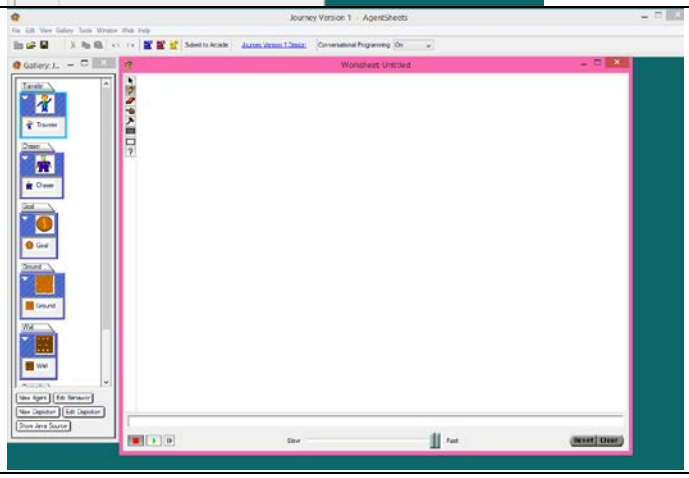
Step 1	**Create Game** **Click on the new game icon (far left)**	
Step 2	**Name the Game** **Name it Maze Craze and click OK**	
Step 3	**Define Agent Size** **Do not change - Click OK**	

Student Handout 1B: Create agents

Step 4	**Create Agent** **Click on New Agent** **Name it Me** **Click ok**	
Step 5	**Edit Agent** **Click Edit Depiction** **Click Clear to erase the current image.**	 *Click on Color> Mask Color>> White to make the white background sections see-through*
Step 6	**Draw Me** **Click Done**	 Here is an example of one way to draw the Me. You can be creative. If you make a mistake, use the eraser or click CLEAR to clear the whole area. Click COLOR>>MASK COLOR>>WHITE to eliminate the white background.
Step 7	**Draw remaining agents**	Floor (yellow box) Wall (Pink box) Goal (Blue Box) Chaser (Green face)

The worksheet is the game space –

it is where the agents will perform their actions.

Step 8	**Make the worksheet** **Click File>>New Worksheet**	
Step 9	**Make the worksheet bigger** **Notice it is big, but not so big that it fills up the whole space.**	

Select Tool

Pencil Tool – places a single agent on the worksheet

Eraser – erases agents from the worksheet

Will be defined later

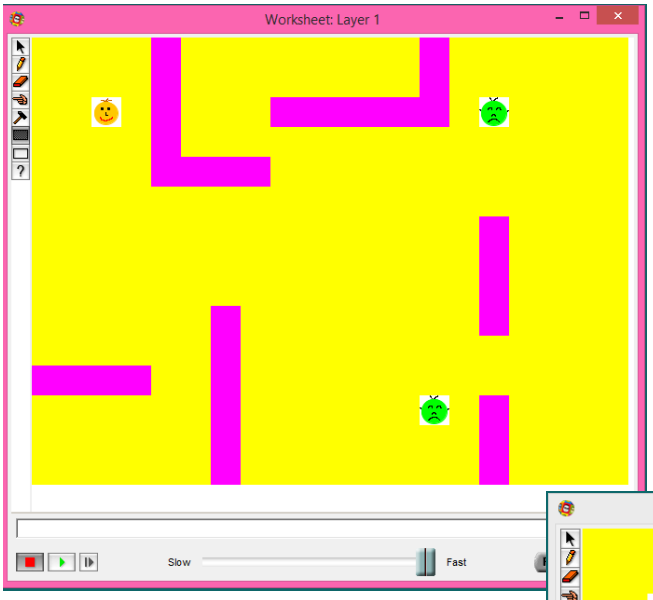
Will be defined later

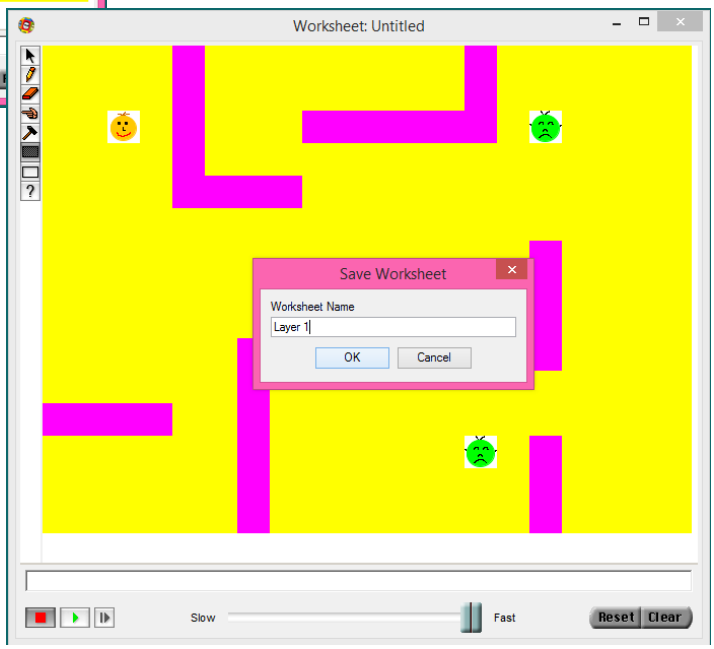
Draw Rectangle – places agents in an array (rectangle)

Erase Rectangle – erases agents in an array

Will be defined later

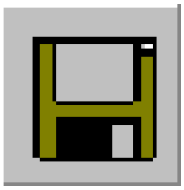
Step 10	**Use the tools to place items on the worksheet.** **Pencil: places agents one at a time** **Filled in Rectangle: Places agents in an array.**	 **Placing one agent on top of another stacks them; however, you can only see the top agent in a stack. Therefore, it is important to use the worksheet construction tools carefully. To use the shaded rectangle tool, click in the upper left corner of the worksheet, click, then drag the cursor to the lower right corner of the worksheet and release. This will produce a single uniform layer of "Background" agents. You can click on the lower right boundary of the worksheet window and stretch it to leave a little "white" space on the right and bottom edges of the "Background" agents, so that you can see where the simulation area ends..**

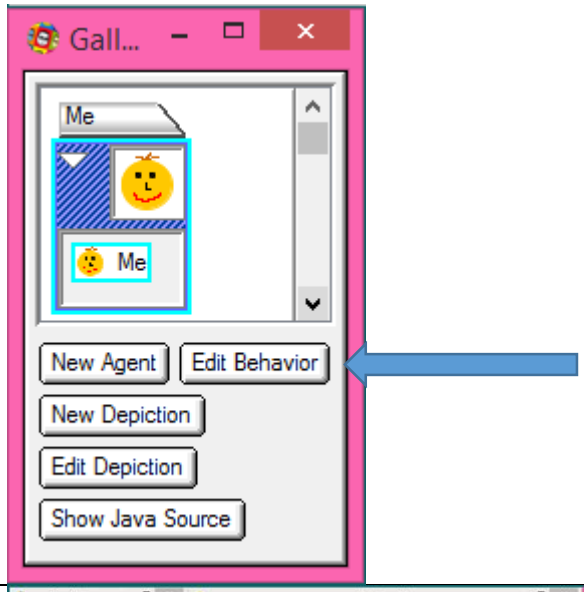
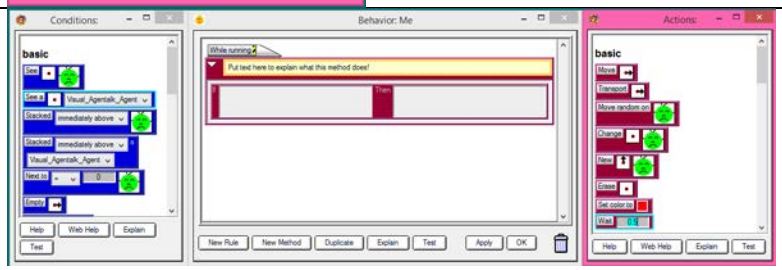


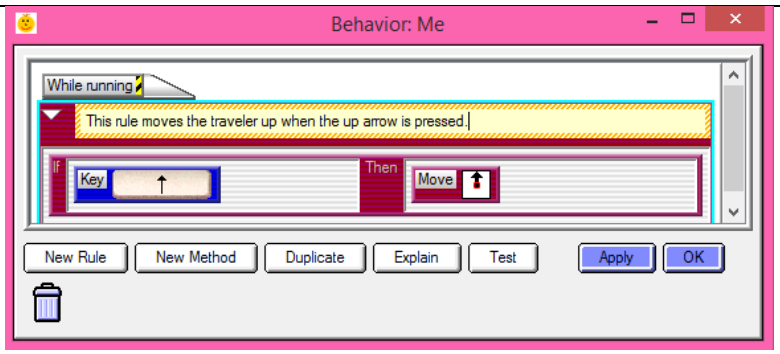
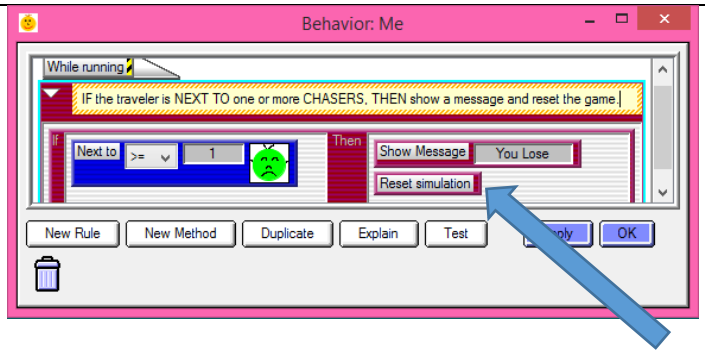
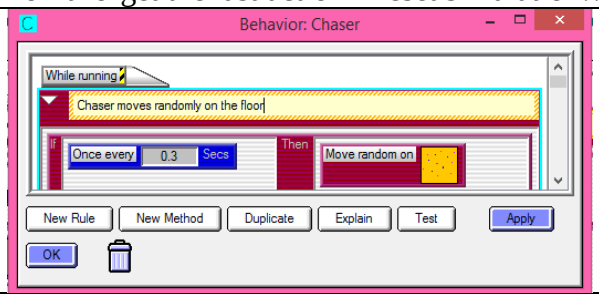


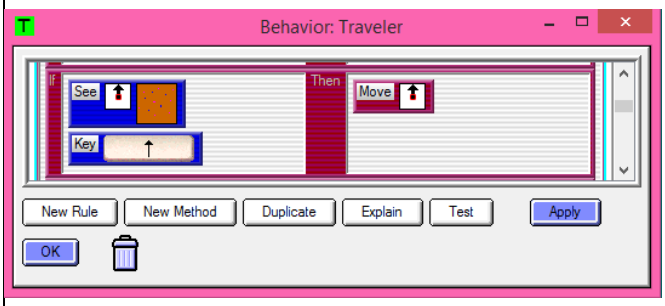
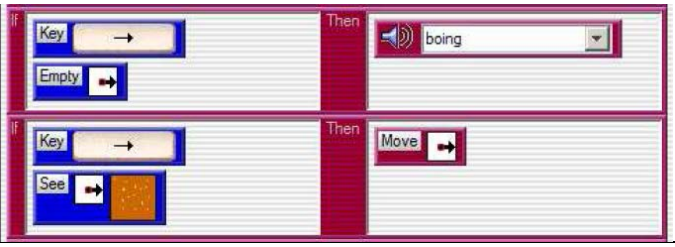
This is a good time to save the worksheet!

File>>Save



Step 11	**Create behaviors for your agents** **Read the explanation** **and then** **Click Edit Behavior**	The kind of behaviors that we will give to our Agents are called rules. Rules are made up of an IF-THEN statement. For controlling the Traveler using the cursor keys, one of the rules we need should be that "IF the Up key is hit, THEN the Traveler will move up." Overall we should have 4 rules, one for each direction (Up, Down, Left, Right). 
Step 12	**Set up your workspace** **When you first open EDIT BEHAVIOR, it will be blank. It is helpful to set up your space in this manner, with conditions (blue) on the left, and actions (red) on the right.**	 **You are going to drag and drop the conditions (on the left) and the actions (on the right) to create the rules.** **To drag and drop the conditions, move the cursor to the solid color so that a hand appears. Then pull it into the empty space of the IF or THEN portion of the rule.**

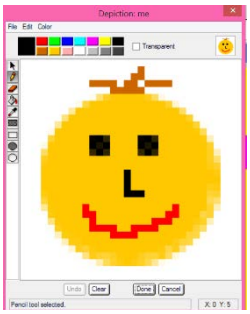
Step 13	**Create a behavior (rules) to make the Traveler (called Me) move using the arrow keys**	 **Take a look at this rule…it says,** **IF I <u>click on the up arrow</u>, THEN <u>my traveler will move UP</u>** **Create the rules to have the traveler move up, right, left and down.** **NOTE: Each rule has to be separate…use NEW RULE to create each new rule.**
Step 14	**Create rule to end the game when the traveler is next to the Chaser** **Click on Traveler and Edit Behavior** **Add these rules**	 Don’t forget the last action – reset simulation!!!
Step 15	**Create rule to end the game when the traveler is next to the Goal**	No hints here – your turn to figure it out. Use step 13 as a hint. Don’t forget the last action – reset simulation!!!
Step 16	**Program the ***<u>Chaser</u>*** to move randomly**	 ***Click on the agent to add behaviors to that agent***

Step 17	**Prevent your Traveler from walking through walls** **Part a)** **Add the code shown** **Part b)** **Add code for the remaining directions**	Work with the person next to you to figure out how to prevent the Traveler from walking into a wall. Here is one way to think about it… Challenge yourselves to find a different way.  ***Click on the agent to add behaviors to that agent*** **Note an important programming point**: The two conditions are in the same box…this is an AND statement. It reads like this: *IF <u>the up arrow is pressed</u> AND <u>the traveller sees ground above him</u>* *THEN <u>he moves up</u>*
Step 18	**Determine where the traveler can cheat**	Traveler can cheat by moving off the game ground. Talk with the person next to you about where this can happen on your worksheet.
Step 19	**Stop the traveler from cheating**	Add rules that make a sound for attempted movement off the ground. Note the importance of rule order for the new rules. Here is an example to prevent the Traveler from moving right off the worksheet. What other direction limit will you need?  ***Click on the agent to add behaviors to that agent***

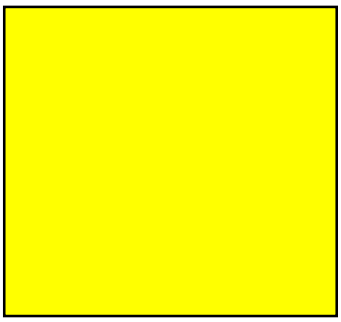
Student Handout 1C:
Agent Creation Models

Use these as quick starting points for your own agent. They don't have to look exactly like the model!

Me



Floor



Wall

Goal

Chaser

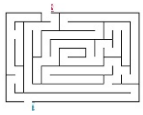
Student Handout 2[2]:

Maze Challenge

Before your start this challenge:

You must be have designed a fully functional Maze Craze Game

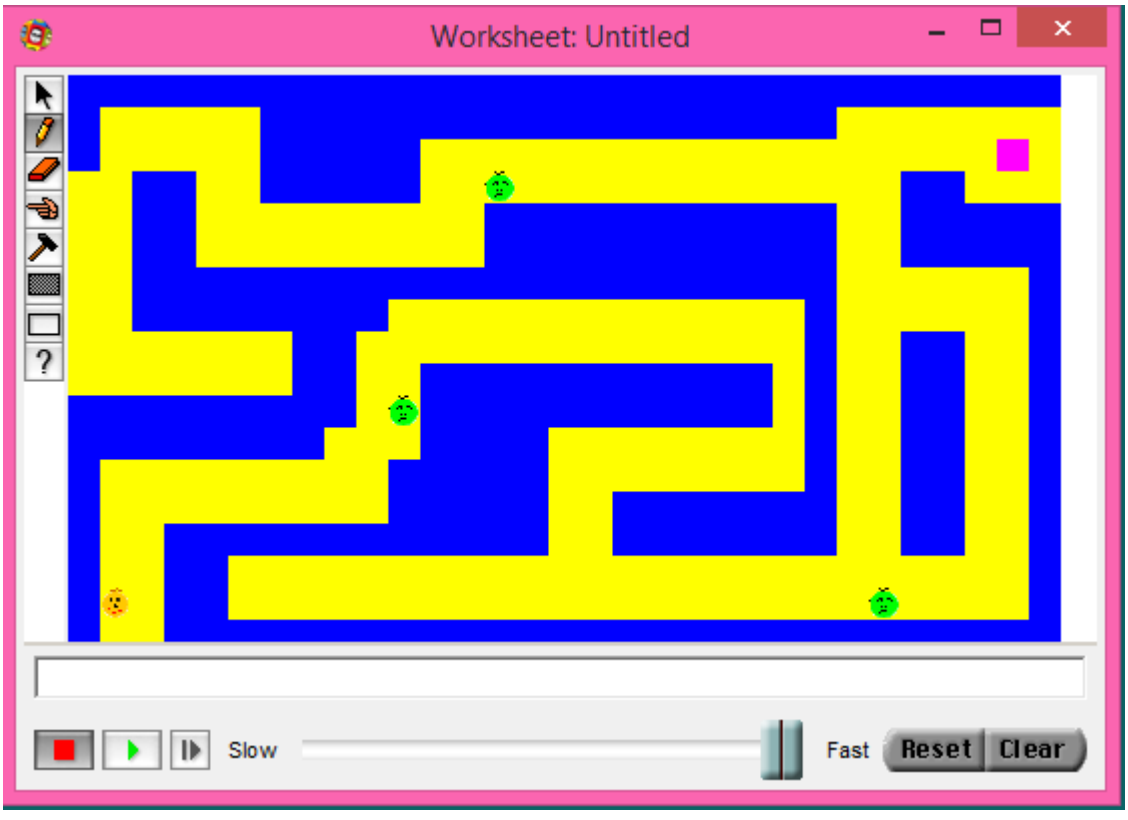
Maze-Crazy!



Give the game a real maze feel!

Design Challenge: By using a smaller agent size, more agents fit on the worksheet, making for more interesting games. Redo Maze Craze using an agent size of 16x16 instead of 32x32.

Redesign Maze Craze with smaller agents and a more maze-life feel. This one is pretty easy – we think you can make it even more challenging!



[2] Included in STANDARD student packet

Maze Craze

ISTE Standards[3] specific to the implementation of Maze Craze (Denoted with (✸))

Creativity and Innovation

Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology. Students:

Apply existing knowledge to generate new ideas, products, or processes:

- ✸ Design and develop games
- Design and develop computational science models

Create original works as a means of personal or group expression.

- ✸ Design original games
- Model your local environment, e.g., ecology, economy

Use models and simulations to explore complete systems and issues.

- Model scientific phenomena, e.g., predator / prey models
- Create visualizations

Identify trends and forecast possibilities.

- Build predictive computational science models, e.g., how the pine beetle destroys the Colorado pine forest
- Build live feeds to scientific web pages (e.g, weather information), process and visualize changing information

Communication and Collaboration

Students use digital media and environments to communicate and work collaboratively, including at a distance, to support individual learning and contribute to the learning of others. Students:

Interact, collaborate, and publish with peers, experts, or others employing a variety of digital environments and media:

- ✸ Students work in teams to build and publish their simulations as web pages containing java applets.

Communicate information and ideas effectively to multiple audiences using a variety of media and formats.

- Effectively combine interactive simulations, text, images in web pages

Develop cultural understanding and global awareness by engaging with learners of other cultures.

- ✸ Students and teachers from the four culturally diverse regions interact with each other

Contribute to project teams to produce original works or solve problems.

---

[3] ISTE Standards for Students (ISTE Standards•S) are the "standards for evaluating the skills and knowledge students need to learn effectively and live productively in an increasingly global and digital world." http://www.iste.org/standards/standards-for-students

Maze Craze (Continued)

- Define project roles and work collaboratively to produce games and simulations

Research and Information Fluency

Students apply digital tools to gather, evaluate, and use information. Students:

Plan strategies to guide inquiry.

Explore web sites and identify interesting connections

Locate, organize, analyze, evaluate, synthesize, and ethically use information from a variety of sources and media.

Find relevant related web-based information, compute derivate information

Evaluate and select information sources and digital tools based on the appropriateness to specific tasks.

Understand validity of information, e.g. Scientific journal information vs. Personal blogs

Process data and report results.

Write programs to access numerical information, define functions to process data and create output based on voice or plotting to represent data.

Critical Thinking, Problem Solving, and Decision Making

Students use critical thinking skills to plan and conduct research, manage projects, solve problems, and make informed decisions using appropriate digital tools and resources. Students:

Identify and define authentic problems and significant questions for investigation.

Define research questions and explore approach of exploration

Plan and manage activities to develop a solution or complete a project.

- Outline sequence of exploratory steps
- Experience complete bottom-up and top-down design processes
- Employ algorithmic thinking for creating programs to solve problems

Collect and analyze data to identify solutions and/or make informed decisions.

Collect data as time series, e.g., collect group size of predator and prey, export time series to excel, explore various types of graph representations, e.g., x(t), y(t) or scatter y=f(x)

Use multiple processes and diverse perspectives to explore alternative solutions.

- Experience and understand design trade-offs, e.g. Bottom-up vs. Top-down

Digital Citizenship

Students understand human, cultural, and societal issues related to technology and practice legal and ethical behavior. Students:

Advocate and practice safe, legal, and responsible use of information and technology.

- Learn how to use tools to locate resources, e.g., images with google image search, but understand copyright issues

Exhibit a positive attitude toward using technology that supports collaboration, learning, and productivity.

- Stay in the flow, where design challenges match design skills

Maze Craze (Continued)

- Experience success through scaffolded game design activities
- Mentor other students

Demonstrate personal responsibility for lifelong learning.

- Explore options of going beyond expected learning goals

Exhibit leadership for digital citizenship.

- In a collaborative setting become a responsible producer of content for diverse audiences

Technology Operations and Concepts

Students demonstrate a sound understanding of technology concepts, systems, and operations. Students:

Understand and use technology systems.

- Know how to organize files and folders, launch and use applications on various platforms

Select and use applications effectively and productively.

- Know how to orchestrate a set of applications to achieve goals, e.g., make game and simulations using Photoshop (art), AgentSheets (programming), and Excel (data analysis).

Troubleshoot systems and applications.

- Debug games and simulations that are not working

Transfer current knowledge to learning of new technologies.

- Reflect on fundamental skills at conceptual level. Explore different tools to achieve similar objectives.