Course Name: Creating a Real World Game **Time Frame:** 5 Days 45 mins each
Unit/Theme: Scratch Multi-Level Maze **Grade Level: 3rd-8th**

CONTENT AND SKILLS	
Concepts/Themes: ● Computer Science ● Programming Concepts	**Key Vocabulary:** ● Sprites ● Blocks ● Loops ● Conditionals
Essential Questions: ● Can students create an interactive maze game? To build an understanding how digital games are created in a real world setting.	
SKILLS	
Content: Students will know… ● How to make their main sprite move ● How to create a gliding sprite ● Design multiple maze backgrounds ● Return to start of level when touching wall or gliding sprite ● Change background when touching level goal ● Add music to their maze ● Use feedback from others to improve their maze	**Process: Students will be able to…** ● Use a design process to follow guidelines. ● Will add different components by following a checklist ● Run activity to check for debugging opportunities. ● Share the final product with class.
ISTE STANDARDS FOR "STUDENTS" (Student are Educators) Educator Standards Used Ex. 4.b. Students will select and use digital tools to plan and manage a design process that considers design constraints and calculated risk.	
● Knowledge Constructor: Make meaningful learning experiences for themselves and others. ○ d: Build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions. ● Innovative Designer: within a design process to identify and solve problems by creating new, useful or imaginative solutions. ○ a: know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems. ○ c: Develop, test and refine prototypes as part of a critical design process. ○ d: exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.	
ISTE STANDARDS FOR EDUCATORS	

Ex. 4.b. Students will select and use digital tools to plan and manage a design process that considers design constraints and calculated risk.

- Collaborator: Dedicate time to collaborate with students to improve practice, discover and share resources and ideas, and solve problems,
    - Dedicate planning time to create authentic learning experiences that leverage technology.
    - Collaborate and co-learn with students to discover and use new digital resources and diagnose and troubleshoot technology issues.
    - Use collaborative tools to expand students' authentic, real-world learning experiences by engaging virtually with experts, teams and students locally and globally.
    - Demonstrate cultural competency when communicating with students and interact with them as co-collaborators in student learning.
- Designer: Design authentic, learner-driven activities and environments that recognize and accommodate learner variability.
    - Use technology to create, adapt and personalized learning experiences that foster independent learning and accommodate learner differences and needs.
    - Design authentic learning activities that align with content area standards and use digital tools and resources to maximize active, deep learning.
    - explore and apply instructional design principles to create innovative digital learning environments that engage and support learning.
- Facilitator: facilitate learning with technology to support student achievement of the 2016 ISTE Standards for Students.
    - Foster a culture where students take ownership of their learning goals and outcomes in both independent and group settings.
    - Manage the use of technology and student learning strategies in digital platforms, virtual environments, hands-on makerspaces or in the field.
    - Create learning opportunities that challenge students to use a design process and computational thinking to innovate and solve problems.
    - Model and nurture creativity and creative expression to communicate ideas, knowledge or connections.
- Analyst: provide alternative ways for students to demonstrate competency and reflect on their learning using technology.
    - provide alternative ways for students to demonstrate competency and reflect on their learning using technology.

Wyoming Computer Science Standards

Computer Systems:
Troubleshooting

- 5.CS.T.01 Identify hardware and software problems that may occur during everyday use, then develop, apply, and explain strategies for solving these problems.

- Practice 6.2 Testing and Refining Computational Artifacts

Algorithms and Programming:
AP.A
PRIORITY
- 5.AP.A.01 Using grade appropriate content and complexity, compare and refine multiple algorithms for the same task and determine which is the most appropriate.
- Practice 3.3 Recognizing and Defining Computational Problems
- Practice 6.3 Testing and Refining Computational Artifacts

AP.C
PRIORITY
- 5.AP.C.01 Using grade appropriate content and complexity, create programs that include sequences, events, loops, and conditionals, both individually and collaboratively.
- Practice 5.2 Creating Computational Artifacts

AP.M
SUPPORTING
- 5.AP.M.01 Using grade appropriate content and complexity, decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
- Practice 3.2 Recognizing and Defining Computational Problems

SUPPORTING
- 5.AP.M.02 Using grade appropriate content and complexity, modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
- Practice 5.3 Creating Computational Artifacts

PRIORITY
- 5.AP.PD.01 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
- Practice 6.2 Testing and Refining Computational Artifacts

SUPPORTING
- 5.AP.PD.02 Using grade appropriate content and complexity, observe intellectual property rights and give appropriate credit when creating or remixing programs.
- Practice 5.2 Creating Computational Artifacts
- Practice 7.3 Communicating About Computing

SUPPORTING
- 5.AP.PD.03 Using grade appropriate content and complexity, test and debug (i.e., identify and fix errors) a program or algorithm to ensure it runs as intended.
- Practice 6.1 & 6.2 Testing and Refining Computational Artifacts

SUPPORTING

- 5.AP.PD.04 Using grade appropriate content and complexity, describe choices made during program development using code comments, presentations, and demonstrations.
- Practice 7.2 Communicating About Computing

(+) ENHANCED

- 5.AP.PD.05 Using grade appropriate content and complexity, with teacher guidance, perform varying roles when collaborating with peers during the design, implementation, and review stages of program development.
- Practice 2.2 Collaborating Around Computing

INSTRUCTIONAL PLAN

List the steps of the lesson, including instructions for the students.

- Day 1: Students will work through the packet of Scratch cards.
- Day 2: FInish Scratch cards. Choose sprites for maze (main, gliding, goal). Design 3 backgrounds.
- Day 3: Program main sprite to move and return to start when touching walls. Program gliding sprite.
- Day 4: Program main sprite to return to home when touching gliding sprite. Program level goal to change backgrounds when touched by the main sprite. Add sounds and other things to make your game more engaging.
- Day 5: Finish programming. Last 20 minutes play 3 people's maze and give verbal feedback about what worked, how to improve, and what you liked. Let 3 people play your game and use feedback to improve.
- Daily: Students can be moving around the room asking each other questions and sharing their code.

ASSESSMENT(S) / PROJECTS / PRODUCTS

- A working 3-level maze.

MATERIALS / RESOURCES

- Scratch Cards
- Scratch
- Project Check-off list

Extra

- Have students code a MicroBit to provide visible feedback
  - Button A - Positive
  - Button B - Needs Improvement