Welcome to the 2017 Marquette University ACM programming competition! For today's competition, you and your team will have 3 hours to develop solutions to several programming problems using the Scratch platform.

Competition Instructions

Problem Templates:

- A few of the problems have optional or required project templates for you to start from when working on your solution.
- Navigate to the template URL: https://scratch.mit.edu/users/MUScratchDivision17/
- You must be signed into Scratch...
    - Click ‘See Inside’ on the project template, then click ‘Remix’ (button in the top-right of the screen)
    - Edit your remix to complete the problem

Submitting solutions, getting feedback, and request clarifications:

- Navigate to the competition URL: http://www.mscs.mu.edu/~cmorley/
    - NOTE: you need to type “www” in front of “mscs.mu.edu” - you will not be able to access the website if you omit it
- Use your given team password to access the competition forms
- To submit:
    - Download your project from scratch with File -> Download To Your Computer
    - Select the problem you are submitting to
    - Upload your .sb2 file (and your .txt design document for creative problems)
    - Hit submit

Routinely check for feedback and clarifications, and let a room proctor know if you are having any problems with the submission system, or suspect your submission did not go through.

Submission Policies

- Try to score as many points as possible. You don't have to attempt every problem.
- Submit your solution as soon as it is ready - don't wait to submit all your solutions at the end. Judges will give feedback on your solutions throughout the competition.
- There is not a penalty for re-submitting to a problem. You can improve your solution according to judge feedback and re-submit to earn more points.
- Do not submit the same solution twice. **Only re-submit to a problem if you have changed your solution.**

There are five problems to solve. Scoring details are as follows:

- Three **Technical Problems** have 5 possible points each to earn, totalling 15 points for the technical section.
    - Points are awarded for how many of the problem requirements were fulfilled, and how well they were fulfilled. Partial credit will be awarded!
- Two **Creative Problems** have a possible 15 points each to earn, but the creative section is capped at 20 total points.
    - On one hand, you could balance your effort between the two problems to try to earn 10 points on each. Alternatively, you could go all-out on one problem to earn 15 points, and only need to earn 5 points on the other to get maximum credit.
    - **For this section, in addition to submitting your Scratch project file, you will need to submit a short "design document" (.txt file from Notepad or Notepad++) listing the features you created for your project.** Since the creative section is very open-ended, this design document lets you "sell" your solution to the judges. These problems can take a lot of work, so brag about your favorite features of your solution - make sure none of them are overlooked!
    - Points are awarded for how well the solution fits the problem's prompt, and the quantity and quality of the features included (and listed in the design document).

Competition Rules

You may use outside sources (your internet, image editing tools, etc.) to create sprites, sounds, and other materials for your project.

You may not take code from public projects on the scratch website, or projects you have worked on outside of the competition.

You may not collaborate with other teams; sharing materials or discussing the problems with other teams is prohibited.

You must ask for permission before using electronic storage or communication devices (phones, flash drives, etc.), or any communication or storage software (email, google drive, etc.).

If you are unsure if a specific outside source is permitted, ask a room proctor! Room proctors may also request you to stop using a specific website or device if they suspect it is being used to break the rules of the competition.

Notify a room proctor if you suspect another team is breaking the rules of the competition.

Practice Problem

To make sure the submission system is working for all teams, please do the following before the competition begins:

- Remix the “Practice Problem” template on the template URL (see the competition instructions)
- Make some changes to the project - add a new sprite, perhaps make it move
- Follow the submission instructions to submit your remix to the problem listed as “Practice”
- Check your team’s feedback page to make sure the judges received your submission (it may take a minute for the judges to give feedback)

Problem 1: Maze Solver (up to 5 points)

Given: Full maze designs with solution directions at each junction, solver sprite with skeleton code for following the maze

Task: The participants are tasked to add code to the maze solver sprite which will cause it to solve the given mazes.

The following features are required:

- The solver should be clearly seen following a correct path through the maze (e.g., it should not just teleport to the finish)
- The solver works autonomously - it should not require assistance from the user
- If the solver touches a maze wall, it should reset to the start point of the maze
- The solver should be able to solve other similar mazes that were not provided to the participants. That is, the solver should be coded in such a way that it would be able to follow any other mazes constructed in a similar way (having the same form of solution directions at each junction).

For this problem, participants must submit the given project template with only the solver sprite's code changed. Solutions which change other aspects of the project will not be graded.

To access the project template click **Problem 1**, click 'See Inside', and then click 'Remix'

Problem 2: Guitar Hero (up to 5 points)
Given: Guitar Hero game background, note sprite

Task: The participants are tasked with making a simple form of the classic game Guitar Hero! The following elements are required:

- Random notes should periodically appear on the right side of the background, along the horizontal lines
- Notes should move along the horizontal lines to the left side of the screen
- A different keyboard key corresponds to each horizontal line
    - Indicate which key corresponds to which line in some way
    - A “hit” occurs when a player presses the correct key while the correct note is touching the target on the right side of the corresponding line
    - A “miss” occurs when a player presses a key while no note is touching the corresponding target, or when a note passes the target without being “hit”
- A different sound should play when the player either scores a hit or takes a miss
- The player loses when five consecutive misses are taken. Here consecutive means five misses occur without a hit occurring anytime between them

Participants are encouraged to use the provided project template, but are permitted to use other sprites, backgrounds or sounds, so long as the required elements are still fulfilled.

To access the project template click **Problem 2**, click ‘See Inside’, and then click ‘Remix’

Problem 3: Basketball (up to 5 points)

Given: Basketball, basket

Task: The participants are tasked with designing a basketball simulator.

The following elements are required:

- The basketball should start in one corner of the screen, waiting for user input
- The user should be able to launch the basketball by providing user input
    - The user should be able to adjust the initial speed and direction of the basketball by providing user input
- The ball should exhibit realistic (or nearly realistic) changes to direction and speed due to gravity
- The ball should not leave the screen
- The ball should be able to bounce off at least one object on the screen
- The player should be able to score points by landing in the basket or hitting some sort of target, with the score displayed on the screen
- The player should be able to reset the basketball to the corner (but not reset the score!) with either a keyboard press or a button on the screen

Participants are encouraged to use the provided project template, but are permitted to use other sprites, backgrounds or sounds, so long as the required elements are still fulfilled.

To access the project template click **Problem 3**, click ‘See Inside’, and then click ‘Remix’

Problem 4: Menu (up to 15 points, and up to a combined 20 points with problem 5)
Note that a project template is not provided for this problem.
Task: Take on your dream of owning a restaurant as you create an interactive restaurant menu! A user should be able to select food choices, but how, when, and what happens next is up to you!

An on-screen waiter might prompt the user with vivid descriptions of the most expensive dishes. Other guests might be seen in the background, entering, leaving, and making conversation. Maybe the chef makes the food to order and you can eat it piece-by-piece. Be creative!

Use this project prompt to show off your ability with the visual design side of the scratch platform. Give the user lots of areas to interact smoothly with the application, and make some unique animations!

Design Document: In addition to your Scratch file, submit a short document (a .txt file from Notepad or Notepad++) describing the features of your project. It doesn't have to be too formal or long - just list the primary ways your user can interact with the project, and describe the features you're most proud of. Pretend you're selling your solution - make sure the judges know about all of the features you spent your time on!

Problem 5: Calculator (up to 15 points, and up to a combined 20 points with problem 4)
Note that a project template is not provided for this problem.
Task: Your cutting-edge software development company has decided that the traditional calculator needs a facelift. Design a complex math machine to help math students across the country!

Your project should serve as a tool for performing math calculations, including the basics (plus, minus, times, divide, square root, etc.), but also some others that might not be seen on the traditional calculator. Pull some ideas from your math classes - you might help the geometry student with calculating the third side of a triangle, or the algebra student with the quadratic equation or graphing. Be creative!

Use this project prompt to show off your ability with the technical side of the scratch platform. Let the user smoothly perform lots of consistent mathematical operations; make them practical, but come up with some complex ideas!

Design Document: In addition to your Scratch file, submit a short document (a .txt file from Notepad or Notepad++) describing the features of your project. It doesn't have to be too formal or long - just list the primary ways your user can interact with the project, and describe the features you're most proud of. Pretend you're selling your solution - make sure the judges know about all of the features you spent your time on!