**Game On** is a [Division C](https://scioly.org/wiki/index.php/Division_C) event for the [2016](https://scioly.org/wiki/index.php/2016), [2017](https://scioly.org/wiki/index.php/2017), and [2018](https://scioly.org/wiki/index.php/2018) seasons. It was held as a trial event at the [2015 National Competition](https://scioly.org/wiki/index.php/University_of_Nebraska_2015), and the [2014 National Competition](https://scioly.org/wiki/index.php/University_of_Central_Florida_2014).

The objective of Game On is to create a game with the program [Scratch](http://scratch.mit.edu/). Scratch is a program designed by MIT Media Labs that allows non-programmers to experiment and play with the basics of programming. It is a visual event-driven language in which you drag and drop blocks of code in order to create scripts.

Overview

At the competition, students are given a theme which they must incorporate into their game. Possible themes (that are not intended for tournament use) are listed as Wave, Fire, Gravity, Silly sports, Frogs, and Newton's Second Law.

In [2018](https://scioly.org/wiki/index.php/2018), the games must also follow a game type. The possible game types are:

* For regional and invitational tournament, Collection, Maze and Avoidnce.
* For state tournament, the regional types and Shooting, Racing and Building.
* For national tournament, any two of the above themes or a two-player game may be used.

Teams of 2 are given 50 minutes to complete their game. The entire competition is conducted offline (in the offline editor). Students are allowed to bring writing instruments, headsets, and a microphone. Other computer programs (such as Photoshop), external resources and pre-constructed game assets are NOT allowed.

**Rubric**

The Game On grading rubric for the 2016-17 season can be found [here](https://www.soinc.org/sites/default/files/uploaded_files/Game_On_Rubric_16-17.pdf), and is explained [here](https://www.soinc.org/sites/default/files/uploaded_files/Game_On_Rubric_Explained.pdf). Because of the open-endedness of this event, there are not many explicit rules or explicit requirements in grading.

Event Advice

Since the Scratch program is free, it is recommended that students experiment with the controls so that they are not learning how to use it at the competition. It is free to download [here](https://scratch.mit.edu/scratch2download/). If students can not use the downloaded offline program, the online editor can be utilized. However, it must be kept in mind that at the competition, students will be using the offline editor. Consider brainstorming game ideas for different themes as practice. Have a team member give you a topic, and time yourself (50 minutes) to complete it.

Practice getting a basic set of code that you will start with or incorporate for different situations. Documentations on the different blocks can be found under [the Scratch Wiki](https://wiki.scratch.mit.edu/wiki/Scratch_Wiki_Home). Practice making sprites move smoothly with user control, making sprites slide across the screen with specific speed, stimulating gravity, turning Sprites, and making one sprite move toward another one; Practice creating the Introduction, Help, and Gameplay screens and adding the buttons between them as quickly as possible. Finally, create a few games for each game type.

As for game ideas, play other games and see what the theme of each game might be. Perhaps that will be the theme the judges choose at the competition! You can also look inside pre-existing projects made by experienced Scratchers (courtesy of Creative Commons) to determine specific mechanics of a project. It also doesn't hurt to ask other users for help in the Scratch community by commenting under projects or posting in [the Scratch Forums](https://scratch.mit.edu/discuss/).

**Tips**

*Remember to practice and include:*

**Introduction Screen.** The introduction screen will be the first thing the judges see, so make it catchy to start with a good first impression. Be sure to include a title and a background relating to the game.

**Help Screen.** Your help screen should clearly explain to the judges how to play the game. Avoid creating a game with very complicated rules because it will be harder for the judges to understand. Many times, these instructions can be concisely written on the introduction screen.

**Scorekeeping.** How points are awarded in your game should not be confusing and should be clear. Clear rules on scorekeeping will make it easier to play the game.

**Code Organization.** The judges will have to understand your code, so try to organize it into sections based on the code's function. It is often difficult to understand someone else's code, so name objects appropriately. Use good practices such as adding comments and organizing large scripts using functions. Organize critical steps in your code using broadcasts.

**Time Management.** Make sure that you complete a working game first before making it look pretty. Form follows function, so finish your coding first! (Time management comes with practice so...PRACTICE!)

**Prioritization.** Prioritize the core game mechanics (a functional game is crucial to a high score) before any cosmetics. The particular order in which you program your game can depend on personal style, so try to practice to see what works best for you. This also goes hand-in-hand with time management since you don't want to spend too long doing a particular task (e.g. customizing sprites, searching for sounds, etc.).

**Testing.** If anything, this is one of the most crucial parts of the coding process. TEST YOUR CODE TO MAKE SURE IT WORKS! Testing is crucial in figuring out syntax errors, fine-tuning variables, setting difficulty levels, etc.

**Cosmetics.** Custom sprites and backgrounds are essential to achieving a higher score. Practice using the Scratch Costume/Background editor in both Bitmap and Vector mode. The editor is limited in capabilities, and combining both modes increase the quality and decrease the production time of the sprite.

*Extra Tip: To test a particular string of blocks without running the whole script, double-click the group. This works for isolated blocks too-- you can reset position, broadcast messages, rotate sprites, etc., by double clicking.*