# Game Programming Test

We need you to code our new super/awesome game: Asteroid Rain! Asteroid Rain is a very simple game where a never/ending sequence of asteroids fall from the top of the screen. The player's objective is to survive by hitting asteroids until your time runs out, which is 60 seconds.

The only visible game objects are the asteroids. Some asteroids should be faster than others, although each asteroid has constant speed (no acceleration). The asteroids appear from the top and fall in a straight line to the bottom.

The player can destroy asteroids by clicking on them. The score is the number of asteroids destroyed, and should be displayed on the screen.

In addition to normal asteroids, sometimes a super asteroid will appear. Super asteroids are larger than normal asteroids, and when you click on them they do not get destroyed.

Instead, they split into two normal asteroids.

The stopwatch should appear in the bottom right corner of the screen. It is an analog display starting at 60 seconds and ticking down to zero.

Whenever an asteroid goes through the bottom of the screen, the player loses a life and the screen flashes red for a short time. The game starts with 5 lives. When the player runs out of lives, or time runs out a "Game Over" screen appears, displaying the score and a button to play again.

# Instructions

* You can choose to make the game for Windows, Mac, iOS or Android. It can be 2D or 3D, there is no preference.
* Your code will ideally be written in C# and use Unity3d.
* Please send us a zip file with the full project, so we can build and run it
* Please specify what Unity version, platform, public software or assets (including version numbers) you used to produce your project. Be sure to include whatever is needed to run the code without any extra steps from our part.
* We will evaluate both the running game and the quality of your code, which should be as clear and readable as possible.
* We will not evaluate the art itself. Any animation, polish or special effects that add value to the user experience will be a bonus.
* Please note, the focus of this exercise is to produce simple, clean and easy to follow code which is robust and easy to maintain. We will be evaluating the code on this basis.