# **INFO6020 – Graphics 2 - Mid-term Exam – Winter 2018**

Instructor: Michael Feeney

## The exam format:

* You may use any resources you feel are necessary to complete the exam, but you are to answer the questions **on your own**. I will be looking for plagiarism (i.e. copying) very carefully. There is *no possible way* that the specific code to answer these questions, or the output to the screen, would be very similar to the look of another student’s code. Remember, this is a test and there are very clear policies about cheating on tests.   
  + <http://www.fanshawec.ca/admissions/registrars-office/policies/cheating-policy>
  + <http://www.fanshawec.ca/sites/default/files/assets/Ombuds/cheating_flowchart.pdf>
* The questions are ***NOT*** of equal weight. There are five (5) pages with six (6) questions
* The answers may be one or a combination of the following:
  + Short answer (in your own words)
  + Snippets of code
  + Complete running solutions
* CLEARLY indicate which answer goes to which question. My suggestion is that you place each answer in its own folder, named “Question\_01”, “Question\_02” and so on (or something equally clear). Another option is to create a Visual Studio solution and add a number of projects – one per question – to it. If I can’t make heads or tails of what question is what, I probably won’t even mark it.
* Place any written answers into a Word, RTF, or text file. Again, *clearly* indicate which question you are answering.
* If you are combining answers (which is likely), please indicate this with a “readme” file or some note (*not* buried in the source code somewhere).
* For applications: if it doesn’t build and run, *it’s like you didn’t answer it*. I’ll correct trivial, obvious problems (like you clearly missed a semicolon, etc.), but you need to be sure that it compiles and/or runs.
* You have until **11:59 PM** on **Monday, March 18th** to submit all your files to the appropriate drop box on Fanshawe Online.   
    
  **NOTE:** Although this may “look and feel” like a project, it isn’t, it’s an **exam**, so there is **no concept of “late marks**”; if you don’t submit your files by 11:59 PM, you don’t get any marks at all. *Don’t Be Late submitting.*

(Also be **SURE** that you are actually submitting the correct files)

* You can reach me through e-mail ([mfeeney@fanshawec.ca](mailto:mfeeney@fanshawec.ca)) or by calling the school.
* There is also a **PlyFiles.7z** file you will need. It’s available on FOL with the mid-term.

## Questions:

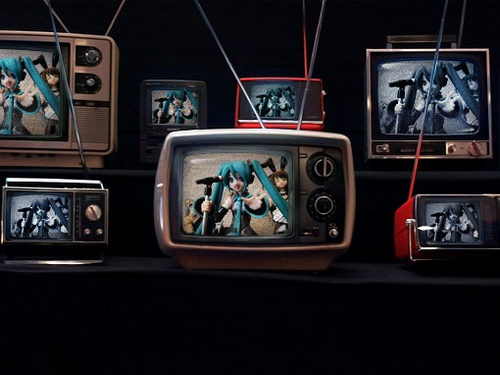
Millions of people watched the Mars “Curiosity” lander touch down.

We will imagine that this happened decades ago, and a lot of people are watching live pictures from the landing site, but through a store window, something like this:



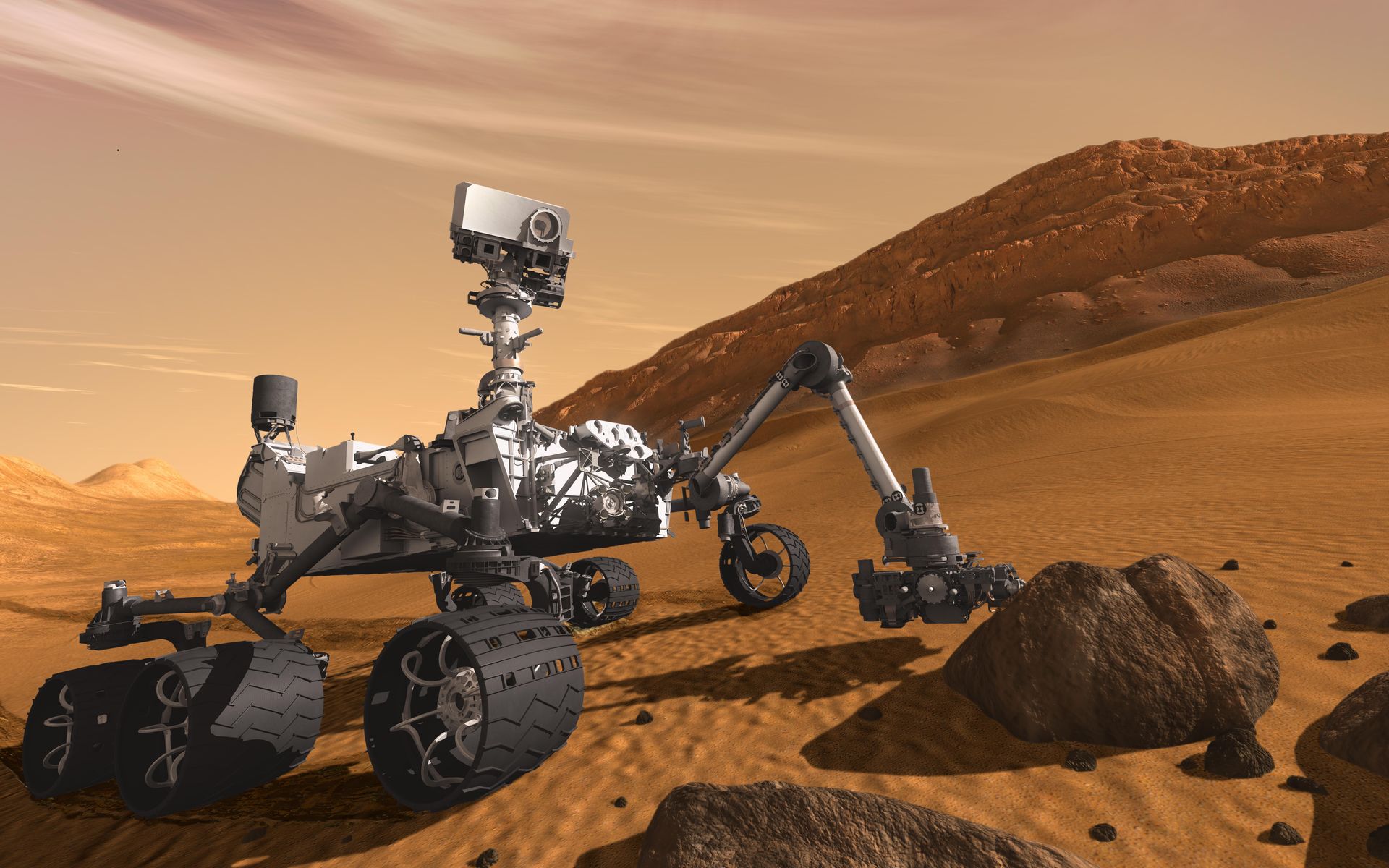
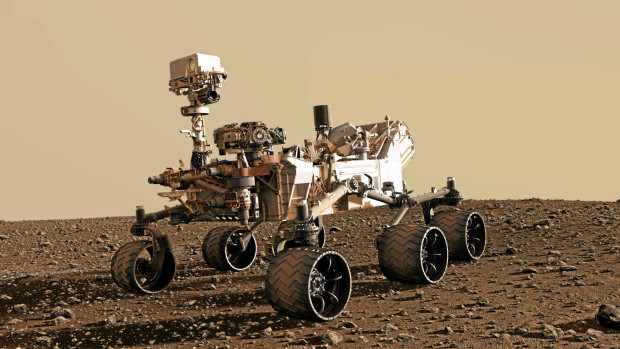
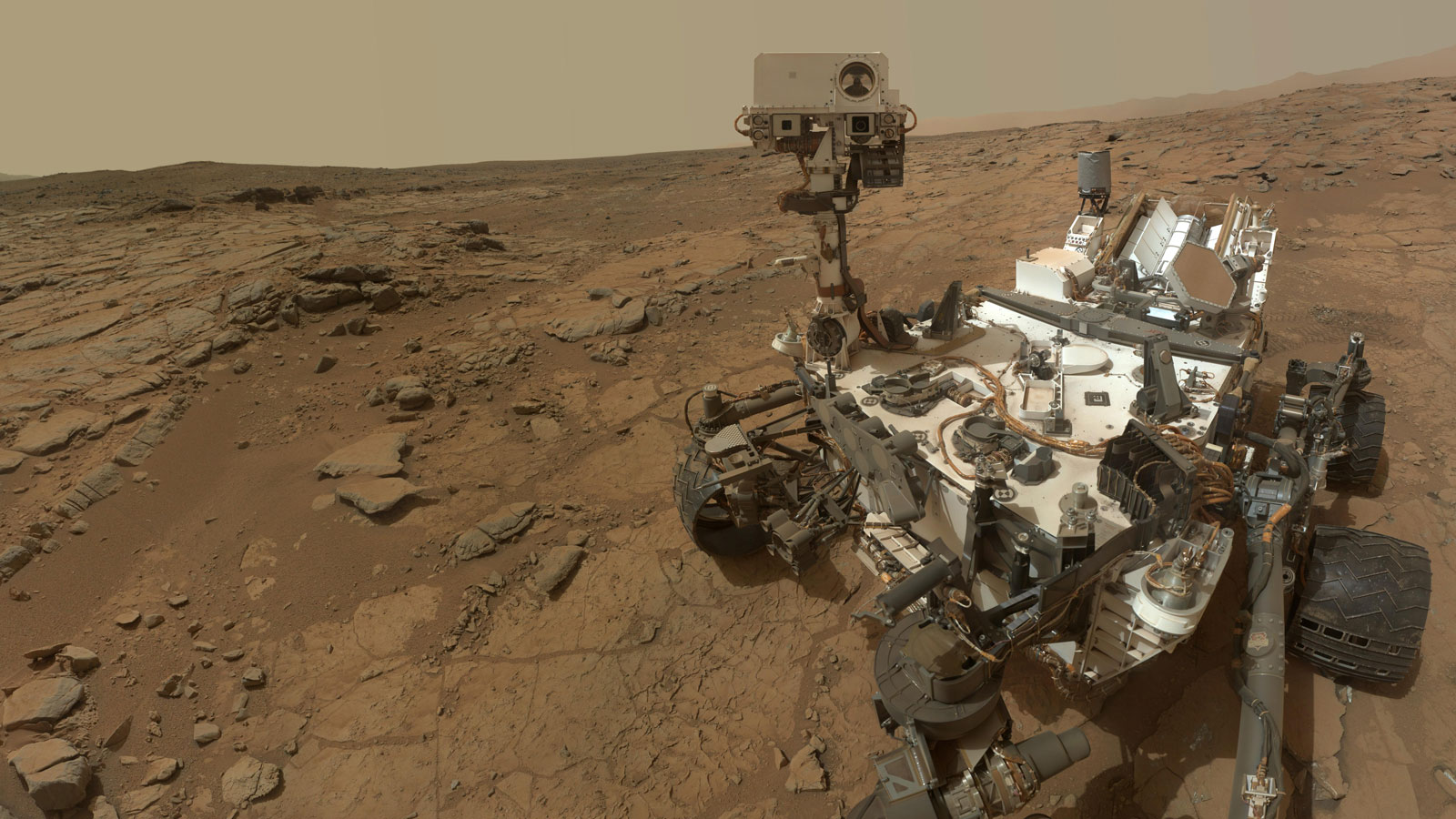
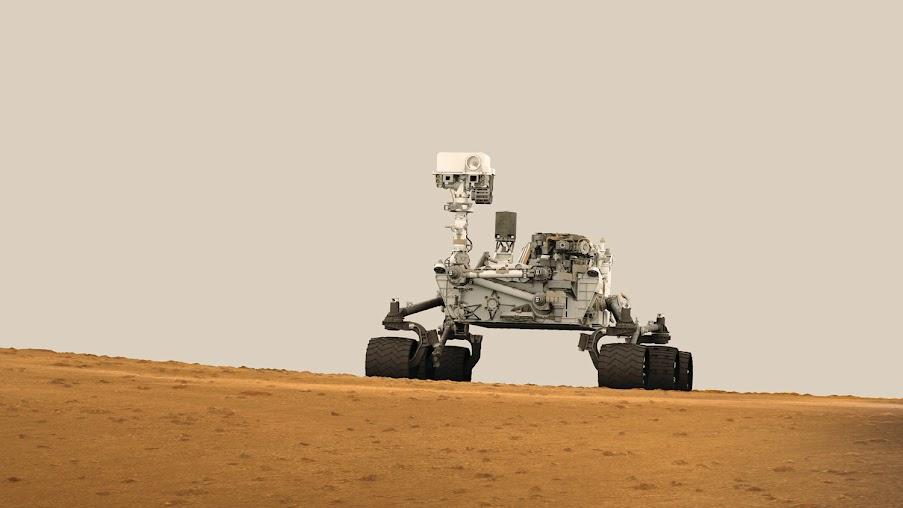
(<https://kaneparker93.files.wordpress.com/2013/09/people-watching-tv-in-window.jpg>)

However, you are to make the image look more like this (with just the old TVs present):

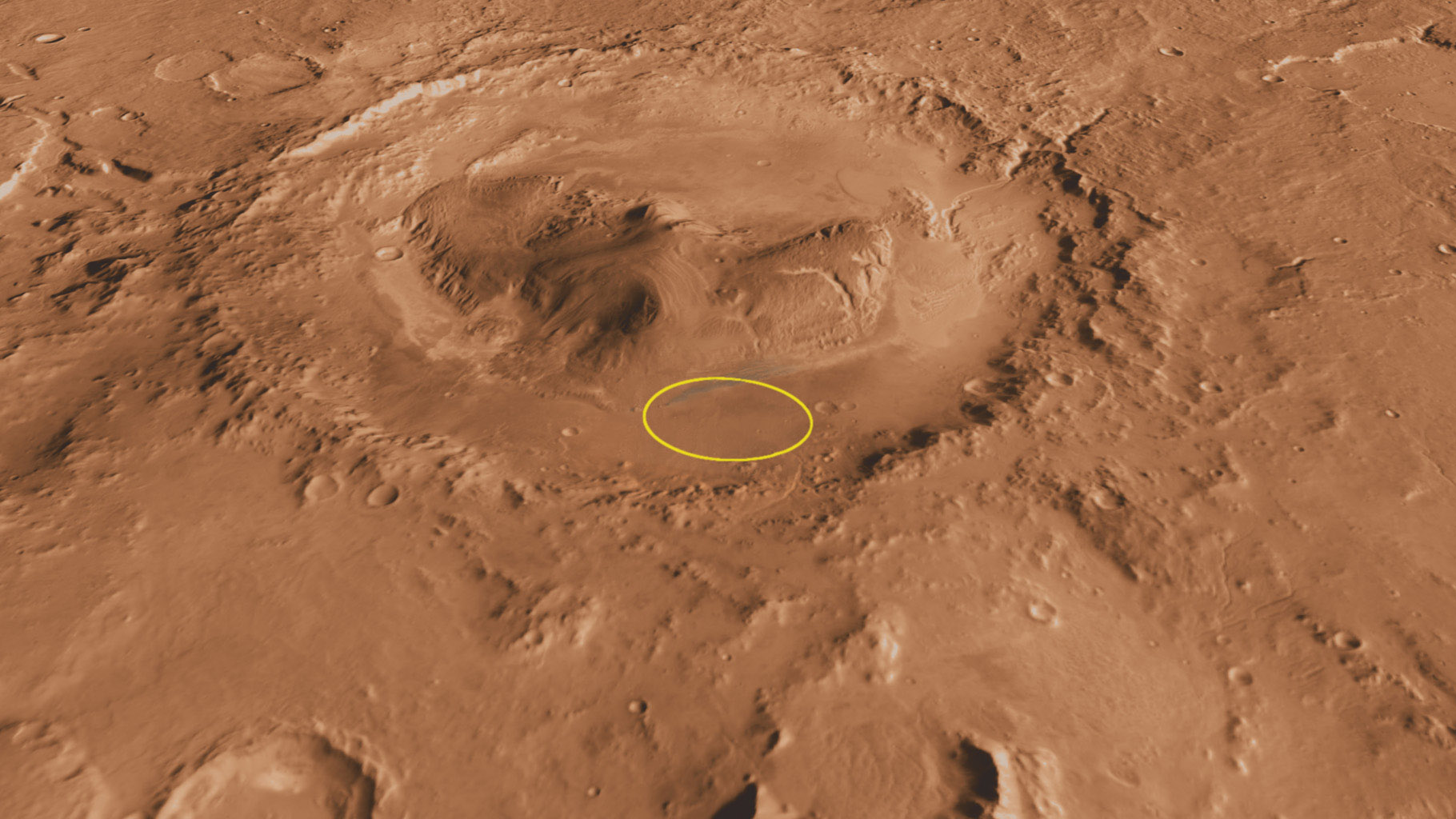


(<https://c1.staticflickr.com/5/4030/4293572142_b6bf9a394f.jpg>)

If you google “Curiosity Rover On Mars”, you will get a number of actual, and “artist rendering” images about what the rover looks like on Mars:



You can also google “Gale Crater”, where the rover landed, to get an idea of what the ground looks like (this picture has a yellow circle showing where they were trying to land the rover – which gives a sense of scale):



1. (10 marks) Place the rover on the Gale crater model. Find a “dirt” texture that is somewhat “Mars like” in colour – which is apparently the colour of rust. Place appropriate lights in the scene so we get an image similar to the ones shown above (evenly lit, daytime). I should be able to clearly see the ground and the rover, and they should be scaled appropriately (like the ground should be MUCH larger than the rover, and I shouldn’t be able to see the edges of the ground model).
2. (20 marks) Alter the scene above, using a camera that is “flying over” the scene, constantly looking at the rover. Imagine if there was one of those small “quad copter” drones, like the ones people on YouTube are always using. For an example, check this out: <https://www.youtube.com/watch?v=dLg484Dq-FM>.   
     
   In this scene, the drone should be flying in a large circle, “orbiting” above and around the rover, and constantly looking at the Rover, something like these shots: <https://www.youtube.com/watch?v=PlSSzUYPy2c>

1. (20) Make a bunch of “cut” shots from each side of the rover, just slightly above. These should change every few seconds.   
     
   So you would show the rover from the front, then from one side, then the back, then the other side, then the front again.   
     
   Unlike the “orbiting” camera in question 2, they would be steady, and after a few seconds, the camera location would instantly switch.
2. (40 marks) Now we get to use the TV model.   
     
   Using a single TV model, place the images from either question 2 or 3, onto the screen. You should only see the TV, yet the image on the screen shows the rover.   
     
   You should be able to move the “game camera” with the keyboard/mouse, which would move around the TV, but the image on the screen should stay the same.  
     
   Note: I’m only going to be moving the camera a little bit (like maybe 30 degrees this way and that, up and down, and move in and out a little) – I’m *not* going to fly all the way around the TV, fly into the screen, or anything like that.
3. (60 marks) Allow the user to “change channels”:  
     
   Using the keyboard, change the image of the TV from question 2 to 3 (and back).   
     
   When the channel changes, briefly add a “static” texture to the image (like old TVs did), for a second.
   * This static should immediately appear when the channel is changed, then “fade” out after a couple seconds.
   * Static is completely random (<https://www.youtube.com/watch?v=8E-MtJBAZvw>), but you can simulate this by:
     1. Having a number of random static textures (at least 6), that you randomly choose each frame, or
     2. (recommended way), randomly offset the texture each frame (add a random number to the U & V coordinates, by passing a random number through a uniform variable, each frame, but use the same, single ‘static’ texture image.
   * The image on screen should be almost completely “static” – but you can still see some of the image
   * After a few seconds, the static fades away, and you can see the rover image
4. (30 mark BONUS) Have 2 (two) TVs in the scene, side by side. You can change the channel on each of them, independently.

(That’s it for the exam).