Memorandum

To:	Shigeru Miyamoto, Game director, and general manager at Nintendo.
From:	[ANONYMIZED]
Subject:	Conflicting Game Goals
Date:	11/2/19

I regret to inform you that some of the goals you mentioned for the new game, *Avoid the Aliens,* will not be possible. There is a conflict with creating the requested game features while also maintaining reduced complexity and reduced computer resource use. You mentioned that you want the game to have different backdrops for each level and unique, disappearing items that give the player points when collected. In the demo version of *Avoid the Aliens,* I made the collectible stars in various colors using the sprite tool. This enabled me to code each star differently so that it would only show up on the corresponding backdrop. I was able to achieve the requested game features, but only at the cost of writing additional lines of code and giving the computer more instructions.

Each star sprite needed to have its own blocks of code in order to perform its job. This included threads that told each star to appear on the correct backdrop, disappear if touched by the player sprite, disappear if it was not collected but the player proceeds to the next level, make a noise when collected, change the "star food" variable by one, and add the star# to the "stars collected" list. There was no way to achieve these game functions with limited statement blocks. I had to use multiple threads for each star sprite. For example, to ensure the stars disappeared at appropriate times, I had to use two threads with different conditional statements and boolean blocks. Thread one stated that when the green flag is clicked, if touching sprite1, change star food by 1 and then hide. The conditional statement in this scenario is the "if, then" block. The boolean block comes from the sensing category and reports a true or false value for "touching sprite1." A second thread was necessary to ensure the star would still disappear at the event of a backdrop change even if it was not collected by sprite1. This thread utilized a boolean block from the variable category, the ">" equation. It stated, when backdrop switches to backdrop#, if backdrop# is greater than the number it is currently on, hide.
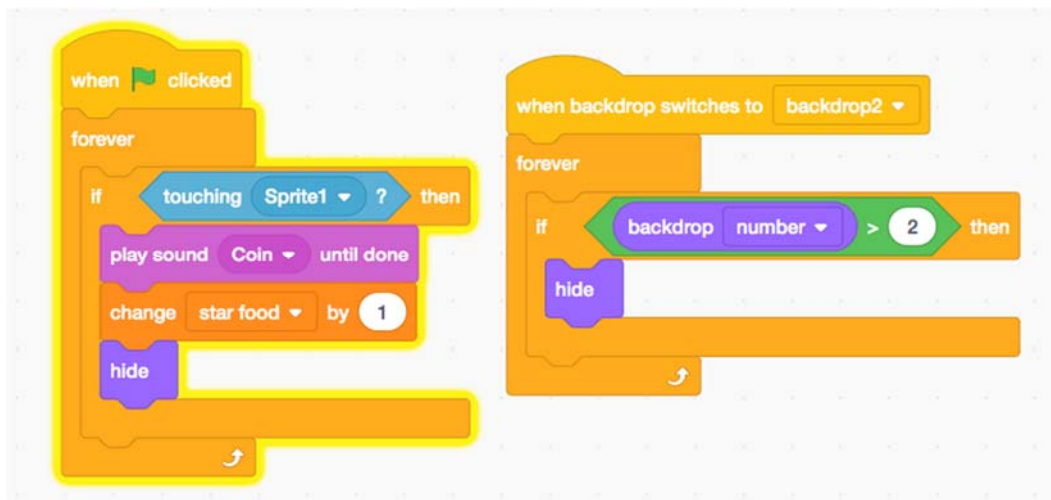
It seems that form and content have a unique relationship in the world of programming. From my experience with this project I found that the form was not extremely limiting to the content. In other situations such as assignment 1, form had the power to heavily affect content. However, in programming, I believe that content is only affected by form if there are multiple conflicting goals you want to achieve. In assignment #3 I did not feel like scratch was restricting the things that I wanted to include in my game. I only felt limited by the fact that I am a beginner with this language and I was trying to write "good" algorithms. Meaning, I tried to ensure that each thread had a specific purpose or reason for existing. There were times when I included code blocks that I then realized were not serving much of a purpose. I then deleted them and tried to achieve the same goal using shorter threads with less code blocks. Overall I would say that the form will most likely allow any one goal to happen, it is only a matter of whether multiple goals conflict and trying to write a "good" algorithm.

The complexity of this scenario does not align with the goals of writing an algorithm with reduced lines of code and reduced computer resource use. The computer needed to receive all of these instructions in order to function in the desired way. Fortunately, today's computers are usually not affected by a few additional instructions so game quality should not be affected. Additional issues might occur in the future of this game if more levels are added. More sprites will be needed and each will have to contain their own threads. This could ultimately present lag issues. An alternative method that I tried to use was cloning the star sprite, however, I was unable to give the unique, specific instructions to each clone which made the game components more difficult to achieve. In addition, cloning would limit the creativity of the game because each clone has the same costumes, sounds, and scripts. It would not be possible to make the collectable items unique to each backdrop if cloning was used. In the case of this game my algorithm successfully achieves the goals that were set forth for the game functions, however, it falls short in reducing resources and complexity. As an experienced game producer, I'm sure you will understand that many desirable goals often conflict with each other in programming.