Review

1. What was a success and what would you recommend for improvement?

**We were extremely happy to have the level making process improved by using the TileD program to design levels. This, after much struggle, was successful and made the process of level design far easier and pleasing. Also, getting collisions working properly, especially with the mirrors was especially difficult. This is because the LibGDX library only allows a small amount of information to be collected about the objects in collision, thus I had to find which specific pellet collided which specific mirror myself. Then I had to send the pellet moving in the correct direction based on the rotation of the mirror.**

**We had many struggles getting simple things working just like the issues stated above. Due to this We tried many different things to solve these problems. Some of this code was left in, and some of it is completely useless. We recommend going through a deep search through the code to remove any unnecessary code that may be left over from the brutal trial and error processes we went through.**

**This code however is not an immediate threat to the speed of the game as it still runs cleanly on my very bad phone (Moto G4 play). It is just ideal to have the code as clean as possible if we were to continue development.**

**Another next step would to implement some sort of tutorial process, so people could learn to play the game themselves.**

**Finally, a main menu could be implemented to improve the player experience.**

1. What would you change based on the peer evaluations?

**Nobody gave any suggestions in the feedback section of the google form. Additionally, everybody selected ‘strongly agree’ for every section. The one outlier being somebody selected ‘agree’ for the bug free option but gave no specific feedback.**

**In short, we were not given any ideas for improvement or change of our program from peer evaluations**

1. Based on the rubric for the program, select a mark for each item, and provide written justification with evidence. Indicate which ICS4U topics you used, in what class, and why.

**Use of ifs, loops, arrays: 5/5 – Since the mark is for the ‘use’ of ifs and loops I think we deserve 5/5 considering there are at the very least 70 if statements and loops in the program.**

**Review: 5/5 – I think we identified our proudest achievements in part A of the review and stated two solid improvements we could make. One being the cleanliness of the code, and the other being an improvement in player experience.**

**Use of ICS4U topics (at least 3 for full marks): 8/8 – Class hierarchies were used (Obstacle, PlayScreen, WorldContactListener), File i/o and exceptions were used (level files, fonts, loading fonts came with a host of exceptions needing to be surrounded with try/catch), 2D arrays were used (the levels were essentially a 2D array of Obstacle objects), GUI’s were used (The entire game is a GUI), searching was used to find objects involved in collision in the WorldContactListener class. That’s five (six if file i/o and exceptions are two different topics) ICS4U topics used in this program. If all we need is 3 for full marks 5 should be enough for full marks too.**

**Follows object-oriented approach: 4/4 – The code has 27 classes. The game is dependant on the fact that the obstacles can work independently and perform actions on their own and interact with each other properly.**

**Check in days (2 marks for may 15): 3/3 – We delivered more than what our plan suggested we would on the first check in day, we met our plans expectations by the second check in day as well. We got full marks on both days anyway already.**

**Features of the program: 8/8 – We made the appearance of the program as pleasing as possible (neither of us are very artistic) so we used simple shapes and simple arrangements as skillfully as possible to make the game astatically pleasing. The animations are smooth (rotation, moving pellets, font ‘bounce’ between levels, objects moving between trackpoints, fading in/out of objects, the toggle bounce on pellet impact). File i/o causes no slowdowns when transitioning between levels. The experience is seamless.**

**Bug-free: 4/4 – No major bugs are present in the program that interfere with the player experience. There are a few trivial things (i.e when a pellet gets caught between two very close obstacles it phases through them a little bit before bouncing). No game breaking bugs what so ever.**

**Plan: not sure/3 – We didn’t hand it in on time. But Mr.Banjevic gave us a second chance after the first check in day to rewrite our plan and hand it in. We did that on time, so I guess 3/3? We completed all our goals on the plan.**

**UML: 3/3 – We already got 3/3 on this.**

**User interface: 3/3 – We’re very happy with how clean the game looks and runs. The interface is simple as the only action the user can take is to click the mouse over obstacles**

**Documentation (Comments, Follows naming conventions): 4/4 – Comments are used to explain important blocks of code, the names of variables and objects (although some are quite long) clearly describe the object/variable and the use of it. Method names are also very clear descriptors of their purpose.**