Functionality: We put implemented functionalities one by one by implementing each properly and going to the next one after we were done with the first. Our program is very robust to keyboard and mouse inputs, responds to changes with in the program itself. We try to catch wrong input methods by just giving error messages in the case, with more time we would have more detailed error messages for mismatch in proper use images.

Design: Our design is very elegant. Each implementation of breaking, powerups and bouncing as described in our Program specification works precisely. With additional implementations of audio and images to makes the game experience feel more proper.

Creativity: Our game isn't purely original. The core functionality of our game works very similarly to Brick Breaker. However, our implementation of powerups adds a new dimension to it. The changes that come with each powerup implementation with timer and audio addition for the flash factor.

Sophistication: Most of the sophistication in our game lies in maintaining smooth animation, proper edge bouncing, timed powerup implementations with proper breaking from bricks and catching from saucers.

Broadness : We used java libraries we hadn't seen in class such as mouse listener, audio, timer action listen and the likes. We used randomization in order to give bonus bricks random position every turn. Used File Input to add and implement images and audio in out project.

Code Quality: We used the same commenting and indentation style seen in project 1 and 2. Named method to work as they are intended to work. drawPowerup would draw a powerup after checking for the necessary conditions.