**Testing Feature Changes:**

The way we chose to test our new feature is by setting up a level with the in-game level editor and creates a stage to test any interactions between our new feature and player.

For our Intense Gold Bomb, we need to test if it’s freezable or damageable. We need to test if the detonation size is larger than original. We need to test player’s status after Gold Bomb got destroyed. With the in-game level editor, it is really convenient for us to test our new feature.

**Testing Affected Subsystems:**

We needed to check if any other subsystems got affected after we modified few files for Intense Gold Bomb, but since there were no new unexpected dependencies showed up, it is not a huge issue to our team. The things we need to check are: looking for objects that depend on explosion, ensure other bad guys are still freezable or damageable, and make sure player status does not switch randomly. An example for affecting subsystems is that; if explosion file is modified, other in game objects which uses the same explosion file might get affected as well.

*/\**

***Closing:***

*Our new feature is called intense gold bomb. We have two options to implement this feature: either to create a new separated class for new feature or modify current subsystems and classes to allow unique behavior on certain objects. We used SAAM analysis to evaluate our two implementations and we chose the second one. In our impact architecture, there are no unexpected dependencies and only minimal subsystems are affected. Hope everybody enjoy our proposed feature.*

*\*/*