**Game Explanation**

It is a first-person puzzle game that uses the colour to apply objects with different effects. There are 7 different colours each with their own effects. Applying multiple colours to an object will generate a new effect. The game is level-based where they must solve a set of puzzles in order to progress through the game. If the player made an error that hinders the completion of a puzzle, then there is a button that resets the whole level. In each level, there are several energy containers in which the player can use to get a colour. The player can shoot a paintball which will colour certain objects.

**Game Mechanics**

**Graphics pipeline and Camera motion**

**Shaders**

Several shaders are used in the making of this game,

* Dissolve shader, which is used to allow the player to progress through the game either done by rendering a pathway or de-rendering a blocked path for the player to traverse through. The shader revolves around the built-in function clip() where it won’t render certain pixels if the specified value is less than 0. The shader uses the local position of the object to identify the centre of the object where the clipping will start and end. Adding a noise texture will randomize which pixels will be rendered and which aren’t, this gives a dissolving look. Float values are used to increase the level of randomness.

* Hologram shader, which is used for aesthetics. The shader revolves around rendering or not rendering colour of a pixel. This is done using a combination of sin, cos, and max function. If the sin or cos function of the world position of the pixel is negative, then using the max function it will assign the value 0 onto the colour of that pixel. To generate the hologram look, the shader will have the culling turned off which allows the colour to be seen in the inside of the object.
* The Fresnel/rim shader, which is also used for aesthetics. This is done by manipulating the colour’s alpha depending on the pixel’s vertex. The shader will display a high alpha value on the rim of the object and that value will gradually decrease the closer the pixels are to the centre of the object.

**Querying and Observational methods**

**Changes after feedback**

**Sourced code/APIs**

**Group Members**