|  |
| --- |
| The Great Poly-Oven! |

The goal of this task is to introduce the idea of **data structures, generic collections**, **inheritance** and **polymorphism.** There are other topics relating to Unity such as world space UI as well.

This task is designed to be difficult and may contain many new concepts that you have not encountered before. Use all the resources that you can! Collaboration is recommended as long as you know why you’re writing the code you’re writing and are submitting individual work.

|  |
| --- |
| Story |

The manager of your bakery have been wanting to improve the efficiency of your store. He sends out a challenge to the employees to see who has great ideas and promises a raise. While scouring the internet, you find an oven called Poly-Oven that makes any types of pastries you want with just a press of a button! You order it and can’t wait to test it out with the customers but unfortunately it came unassembled.

|  |
| --- |
| Description |

**The character**

The player is in first person mode and you should be able to move your screen with your mouse while being able to click objects in the center of your screen. You are allowed to use any pre-built assets either by you or Unity for this character.

**The pastries**

There are three types of pastry: Cake, cookie and ice cream. Each padstry has a name(**“cake”, “cookie”** and **“iceCream”**) and a sweetness level. Please make sure that the name in your game matches the case of the pastry names. Otherwise, the tester will not work! Each pastry has a sweetness level of 1, 2, and 3 respectively. Represent these pastries as any model in-game as you would like as long as they accurately correspond to their code counterparts.

**The counter and the oven**

Create a scene that contains a counter and an oven. The oven is the place where the baker can choose what pastry to make. There should be a total of 6 buttons to press. There are three that will be for each pastry, one for the special baking technique, one for clearing your current selection (or by pressing ‘X’), and the last one is to bake the pastry chosen. Make sure that some sort of legend exists so that a player can identify what pastry they are currently baking.

The counter will be the location where the pastries are going to be once they’re done “baking” (spawning). It has three spots that are filled one by one. The counter has 1 button that sends the order to the customer.

**Baking**

Each of the button in-game should be a physical button in the world which you can click with the mouse. Here is how the process goes:

1. You press one of the three buttons to choose the pastry. The chosen pastry shows up on top of the oven.
2. If needed, you press the special baking technique button to enhance the pastry. The pastry on top of the oven gets modified
3. If the current setup is not correct, press the clear button or press ‘X’. Removes the pastry and any special baking technique applied to it.
4. You press the “bake” button that will move the pastry on top of the counter. (In this universe, an ice cream can be baked.)
5. When the pastry is done baking, the top of the oven is cleared.

You should not be able to bake another pastry while the current one is baking.

**Special baking technique**

Each pastry can be modified with a special technique. Special baking adds 1 sweetness level to the original value of each pastry. So a cake will have a sweetness level of 2 instead of 1 and so forth. The cake can have frosting on top of it (add a white square on top of your cake object). The cookie can have peanuts in it (add a brown square on top of your cookie object). The ice cream can have an extra scoop (add the same ice cream object on top of the current one). Each special baking technique can only be applied once. Pressing the special baking technique button does not do anything if there are no pastry chosen.

**Sending order**

Pressing the send button in the counter will run the check in-game. If your baking matches the order of the customer requests then there will be a UI that pops up to confirm your order with a message of “Good job!”. Otherwise the pastries on top of the counter gets cleared and you have to start over!

**How to Implement! (Please Read)**

Make a public structure (*struct*) called **pastryInfo** with a field *name* as a string and *sweetnessLevel* as an integer.

|  |
| --- |
| public struct pastryInfo  {  public string name;  public int sweetnessLevel;  } |

A struct called pastryInfo

Once you’ve created this structure you can now call it in other classes like a normal type, remember types are like *int*, *bool,* *float*.

|  |
| --- |
| ...  int variableName;  pastryInfo variableName;  ... |

**Possible Errors**

pastryInfo is missing: make sure you move where you declared the pastryInfo structure outside of any class.

|  |
| --- |
| Information [top](#no3l4bce5ba2) |

**struct (Structure)**

A struct is a custom type that can be used to package groups of related variables together. There is a reference [here](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/struct).