Notes

1. According to the design, there should be no collision so you could push to main branch. (However, strongly encouraged to use branch if you know the correct way to use it)
2. Make sure to read descriptions of all sections to have comprehensive understanding. (Especially interface)
3. Player selection only appears at the end of dialog. (No more dialog after player made selection, for easy implementation)

Interface (Tingwei)

Which feature to implement:

1. **Scrolling itembar:** on left side which can accept item from dialog box's selection.
   e.g. click [accept chocolate] in dialog box -> show chocolate in itembar
2. **Dialog box:** at the bottom of the screen only visible when collision happens.
   e.g. player walks toward the door and the dialog appears: "the door is locked"
   e.g. player walks toward a machine and dialog appears: "what would you like to put into the machine? 1. Banana 2. Chocolate 3. skull" (clickable)
   e.g. player walks toward the closet and dialog appears: "would you like to open it? 1. Yes 2. No"

Which class to implement:

1. **Itembar**
   Fields
      Item[] items (getter, setter)
   Methods
      void DeleteItem(int index)
      void AddItem(Collectible item)

2. **DialogBox**
   Fields
      String[] dialog(setter)
      Itembar itembar
      bool isSelect(setter)

String[] selections(setter)

Methods

void ShowDialog(): show dialog at the bottom, player click to read next

int CloseDialog(Collectible item):

if (isSelect) {show selections at the bottom right of the dialog}

destroy/deactive dialog

if (item!=null) {itembar.AddItem(item)}

return user's selection

Movement/physics (Huiling)

Which class to implement:

1. **PlayerController**

   Fields

   ItemBar, DialogBox

   Methods

   Update(): move in 4 directions (command pattern is not necessary but you can try:)

   OnCollisionEnter2D(Collision2D collision)

   If (collision.gameObject.tag == narrative)

   collision.GetComponent<Item>().Collided();

2. **TeleportPlayer** (handle door collision i.e. change room, similar to TeleportCaptain)

   Fields

   DestinationRoom, MainCamera

   Methods

   OnCollisionEnter2D(Collision2D collision): send player to destination room.

Input (Keying)

Which class to implement:

1. **Item**(abstract class)

   Methods

   Collided(): show dialog, etc.

2. **Collectible**(interface)

   Properties

   All properties required for generating in Itembar

3. **A~I** classes, inherits Collectible and extends Item (Fields: Dialog, Player)

   Name as you like, see https://docs.google.com/document/d/1ILDUVhJNJka4xzWnBZE5qMYnSzHGu30FobBDSL4MjEw/edit#heading=h.7zpb0uke0erx

4. **(a)~(y)** classes, extends Item (Fields: Dialog, Player)

   Name as you like, see https://docs.google.com/document/d/1ILDUVhJNJka4xzWnBZE5qMYnSzHGu30FobBDSL4MjEw/edit#heading=h.7zpb0uke0erx

Animation & visuals (Huiling)

Which UI to implement:

Itembar, Dialog window, Rooms, prefabs for items (A~I, (a)~(y)), attach correct scripts correspond.

Which class to implement:

1. **ItemCameraFollow**

   Basically the same as exercise 1, but make sure camera can only see the current room