```
    private void LetGoOfMouse()
    {
        ActivatedFollowMouse = false;

        // check whether I'm currently in the field where letting go would equip the piece
        SkinsMouseController.Instance.EquipSkinPiece(CurrentlyHeldSkinPiece);

        // remove the object
        Destroy(CurrentlyHeldObject);
        CurrentlyHeldSkinPiece.Data.MyBodyType = Type_Body.None;
        CurrentlyHeldSkinPiece.Data.MySkinType = Type_Skin.None;

        this.enabled = false;
    }

    private void FollowMouseCalculations()
    {
        _mousePosition = Input.mousePosition;
        _mouseWorldPosXY = Camera.main.ScreenToWorldPoint(_mousePosition);

        CurrentlyHeldObject.transform.position = _mouseWorldPosXY;

        if (_mouseWorldPosXY.y > 0)
        {
            if (Physics.Raycast(CurrentlyHeldObject.transform.position, Camera.main.transform.forward, out _hit, Mathf.Infinity,
LayersToCastOn))
            {
                _mouseWorldPositionXYZ = _hit.point;
                CurrentlyHeldObject.transform.position = _mouseWorldPositionXYZ;
            }
        }
        else
        {
            if (Physics.Raycast(CurrentlyHeldObject.transform.position, -Camera.main.transform.forward, out _hit, Mathf.Infinity,
LayersToCastOn))
            {
                _mouseWorldPositionXYZ = _hit.point;
                CurrentlyHeldObject.transform.position = _mouseWorldPositionXYZ;
            }
        }
    }

    public void ClickedSkinPieceButton(GameObject objectToSpawn, SkinPieceElement skinPieceElement, Vector3 spawnPosition)
    {
        CurrentlyHeldObject = Instantiate(objectToSpawn, PanelToInstantiateClosetImagesOn.transform);
        CurrentlyHeldObject.transform.position = spawnPosition;

        CurrentlyHeldSkinPiece.Data.MyBodyType = skinPieceElement.Data.MyBodyType;
        CurrentlyHeldSkinPiece.Data.MySkinType = skinPieceElement.Data.MySkinType;
        CurrentlyHeldSkinPiece.HidesSirMouseGeometry = skinPieceElement.HidesSirMouseGeometry;
        CurrentlyHeldSkinPiece.ScoreValue = skinPieceElement.ScoreValue;

        ActivatedFollowMouse = true;

        this.enabled = true;
    }

    // called on GiveReward() in RewardController
    public void AddNotificationToList(ButtonSkinPiece buttonSkinPiece)
    {
        bool foundNotification = false;

        if (ButtonsWithNotifications.Contains(buttonSkinPiece) == false)
        {
            if (buttonSkinPiece.HasBeenNotified == false)
            {
                ButtonsWithNotifications.Add(buttonSkinPiece);
                foundNotification = true;
            }
        }

        if (foundNotification == true)
        {
            PageController.Instance.ButtonBackpackSuper.IhaveNotificationsLeftCloset = true;
            PageController.Instance.NotifyBackpackSuper();
        }
    }
```

```
79
80      public void NotificationActivater()
81      {
82          for (int i = 0; i< ButtonsWithNotifications.Count; i++) // looping over the list of buttons with notifications
83          {
84              // only do the following if List[i] NotifObject is not ON
85              if (ButtonsWithNotifications[i].NotificationObject.activeSelf == false)
86              {
87                  // activate notif on button skinpiece
88                  ButtonsWithNotifications[i].NotificationObject.SetActive(true);
89                  ButtonsWithNotifications[i].HasBeenNotified = true;
90
91                  // figure out what pager has a similar BodyType to the ButtonWithNotif...
92                  for (int j = 0; j < ButtonsClosetPagers.Count; j++)
93                  {
94                      if (ButtonsWithNotifications[i].MySkinPieceElement.Data.MyBodyType == ButtonsClosetPagers[j].BodyType)
95                      {
96                          // activate notif on found button
97                          ButtonsClosetPagers[j].NotificationObject.SetActive(true);
98
99                          if (ButtonsClosetPagerWithNotifs.Contains(ButtonsClosetPagers[j]) == false)
100                         {
101                             ButtonsClosetPagerWithNotifs.Add(ButtonsClosetPagers[j]);
102                         }
103                         ButtonsClosetPagers[j].IHaveButtonsWithNotificationOn = true;
104                         ButtonsClosetPagers[j].ButtonsWithNotifsOnOnMyPage.Add(ButtonsWithNotifications[i]);
105
106                         OnSkinpieceUnlocked?.Invoke(ButtonsClosetPagers[j]);
107                         break;
108                     } // below statement should add the RIGHT limbs
109                     else if ((ButtonsWithNotifications[i].MySkinPieceElement.Data.MyBodyType == Type_Body.FootRight &&
    ButtonsClosetPagers[j].BodyType == Type_Body.FootLeft) ||
110                          (ButtonsWithNotifications[i].MySkinPieceElement.Data.MyBodyType == Type_Body.LegRightLower &&
    ButtonsClosetPagers[j].BodyType == Type_Body.LegLeftLower) ||
111                          (ButtonsWithNotifications[i].MySkinPieceElement.Data.MyBodyType == Type_Body.ArmRightLower &&
    ButtonsClosetPagers[j].BodyType == Type_Body.ArmLeftLower))
112                     {
113                         // activate notif on found button
114                         ButtonsClosetPagers[j].NotificationObject.SetActive(true);
115
116                         if (ButtonsClosetPagerWithNotifs.Contains(ButtonsClosetPagers[j]) == false)
117                         {
118                             ButtonsClosetPagerWithNotifs.Add(ButtonsClosetPagers[j]);
119                         }
120                         ButtonsClosetPagers[j].IHaveButtonsWithNotificationOn = true;
121                         ButtonsClosetPagers[j].ButtonsWithNotifsOnOnMyPage.Add(ButtonsWithNotifications[i]);
122
123                         OnSkinpieceUnlocked?.Invoke(ButtonsClosetPagers[j]);
124                         break;
125                     }
126                 }
127             }
128         }
129
130         // activate notif on closet button
131         if (ButtonsWithNotifications.Count > 0)
132         {
133             ButtonCloset.NotificationObject.SetActive(true);
134             ButtonCloset.IhaveNotificationsReadyInTheCloset = true;
135         }
136     }
137
138
139
140
```