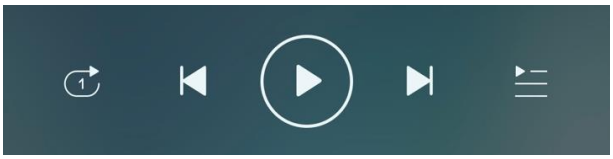
Introduction:

Music Player App is a commonly used cell phone software in our daily life. We often perform interactive behaviors on the user interface when we listen to music on our cell phones.

In Assignment 1, in order to provide smart phone users with a better interaction experience of using music software, I chose the virtual keys of the music player interface as the research object. My approach is to optimize the buttons' triggering and feedback the user experience more concise and effective in preventing false touches. In this task, I will use Bayesian optimization to optimize the selection of these modes using user feedback.

Task design:

The interface of the music player app is shown below, and users can operate the music playback mode through these five virtual buttons:



In assignment1, I assigned each of the 5 virtual buttons on the UI the following interaction properties:

Trigger mode: click trigger, long press trigger

Feedback mode: mute feedback, vibration feedback

For each of the five virtual keys, there will be two sets of interaction properties, one for trigger mode and the other for feedback mode. The interaction properties of these four UI elements affect the user experience to a certain extent.

My task requires improving the combinatorial optimization method in A1 to optimize the interaction properties of these five virtual buttons, which needs to take into account the design of the objective function and how to use user feedback for optimization. Based on the task design and code of A1, I introduced Bayesian optimization into the optimization process.

For the whole optimization process, there are three main aspects:

1. using a multi-objective function to evaluate the button combinations, taking into account the utilization rate and the false-touch rate.
2. for each weight value proposed by the Bayesian optimizer, use the exhaustive search method to search for the best button combination, considering two states (trigger and feedback).
3. Use the Bayesian optimizer to update the weight values through user feedback. The optimal weight values for the two objective functions are found through several iterations.

The details are shown below:

Two Objective functions:

Objective function 1:

In this study, I selected a design goal of maximizing the simplicity of use for the user, which

means that the user will not be bothered by the tediousness of interacting with these five virtual buttons.

We can assume that from the user's point of view, for one of the virtual buttons, it is more convenient to choose the click trigger than the long-press trigger, so when the UI element uses the click trigger method, the system automatically adds one point to the combination of this set of interaction attributes. In addition, if the vibration feedback method is chosen, this is not considered an advantage for the design goal of using simplicity, so no points will be added.

In addition, for the design goal of improving the smoothness of use, the choice of the mode of click triggering has a more direct impact on the experience of the operation than the feedback. So I've given 2 points for the click trigger mode, and only 1 point for the mute feedback. The details are as follows:

Defining X_i as the score for the trigger mode, the scores are shown in the table below:

Trigger mode	Click to trigger	Long press to trigger
X_i	2	0

Defining Y_i as the score for the feedback mode, the scores would be as shown in the table below:

Feedback mode	Silent Feedback	Vibration Feedback
Y_i	1	0

Considering the above system, I set the objective function as below:

$$Max \sum_{i=1}^{5} \left(X_i + Y_i \right)$$

Where "i" stands for the corresponding UI element, as mentioned above there are 5 elements in total. The purpose of this objective function is to select the combination of interaction properties that are best suited to the design objective of maximize the simplicity of the user's use.

Objective function 2:

In order to make the design objective richer and make the design more rational, I added an extra design objective. Considering that the usage rate between these five virtual buttons is different and the functions corresponding to the five virtual buttons are different, I decided to improve these five UI elements from the perspective of reducing the misuse rate.

Considering the two triggering methods and the two feedback methods, I similarly evaluated each of the four interaction attributes under the design goal of minimizing the misuse rate. The click trigger method is prone to misoperation due to its short duration as compared to the long-press

method. Therefore, the long-press type is better able to fulfill this requirement. Vibration feedback is more likely to let the user realize that they have operated the five virtual keys of the UI than mute feedback, so that they can realize whether they have made any misuse, and thus further avoid the misuse generated by the user unconsciously. Therefore, vibration feedback also better meets the design goal of minimizing the rate of user misuse.

In the objective function of minimizing the user misuse possibility, the assignment of points is also used. If clicking is used, no point is added, and if silent feedback is used, no point is added. For the design goal of reducing the possibility of user misuse, the long-press trigger mode of operation can truly prevent users from misuse. The vibration feedback can only remind the user that they have just performed an operation, and is more of a warning and reminder function. So I gave 2 points for the long press trigger mode, and only 1 point for the vibration feedback.

In the design objective of minimizing the user misoperation rate, the objective function is as follows, x_i denotes the score of the ith button in the trigger mode, and y_i denotes the score of the ith button in the feedback mode. The details are as follows:

Defining x_i as the score for the trigger mode, the scores are shown in the table below:

Trigger mode	Click to trigger	Long press to trigger
x_i	0	2

Defining y_i as the score for the feedback mode, the scores would be as shown in the table below:

Feedback mode	Silent Feedback	Vibration Feedback
y_i	0	1

Considering the above system, the objective function as below:

$$Max \sum_{i=1}^{5} \left(x_i + y_i \right)$$

So, in objective function one: the trigger method is click can add one point; When the feedback method is mute can get one point. In objective function two: the trigger mode is long press adding 1 point; the feedback is vibration adding 1 point.

Then, I form a Multi-objective function by combining these two objective functions. This multi-objective function is applied to the combinatorial optimizer to achieve Multi-objective optimization. Considering that these two objective functions conflict with each other, I assign weights to them.

For users, the attribute of ease of use of high usage buttons is more important than the attribute of

preventing false touches. The opposite is true for low-usage keys. Considering that these five keys have different usage rates, I decided to use the key usage rate as the weight of each key score in the objective function 1. And the difference between the number 1 and the utilization rate of that key was used as the weight for the score of that key attribute in the objective function 2. The output of the multi-objective function is used to evaluate the combination of keystroke properties. The new multi-objective function is as follows:

$$Max\begin{Bmatrix} W_1\times\left[F_1\left(X_1+Y_1\right)+F_2\left(X_2+Y_2\right)+F_3\left(X_3+Y_3\right)+F_4\left(X_4+Y_4\right)+F_5\left(X_5+Y_5\right)\right]+ \\ W_2\times\left[\left(1-F_1\right)\times\left(x_1+y_1\right)+\left(1-F_2\right)\times\left(x_2+y_2\right)+\left(1-F_3\right)\times\left(x_3+y_3\right)+\left(1-F_4\right)\times\left(x_4+y_4\right)+\left(1-F_5\right)\times\left(x_5+y_5\right)\right] \end{Bmatrix}$$

Where W_1 and W_2 are the weights of objective function 1 and objective function 2, respectively, and the magnitude of their values determines whether the objective function is more or less important in the combined objective function. F_1 to F_5 are the utilization rates of the keys respectively.

Approach to solver:

When we define a multi-objective objective function, we need to evaluate all possible button combinations using that multi-objective function to get the best button combination under that objective function. In this design, we use the exhaustive enumeration method to optimize the button configuration of the music player interface. The exhaustive enumeration method ensures that the global optimal solution is found in a limited and well-defined search space.

In this task, each button has two trigger states and two feedback states, generating a total of 2^{10}=1024 possible combinations. A search space of this size can yield accurate answers very quickly on current Colab computational resources. Due to its comprehensiveness, this approach is guaranteed to find the best configuration by evaluating every possible combination, avoiding the risk of missing the optimal solution. In the current task environment, this solution is very efficient and accurate.

User feedback and Bayesian Optimizer:

In this project, the combination of user feedback and Bayesian optimizer is the key part. The core of this approach lies in learning from actual user feedback and optimizing design decisions. Users are asked to give ratings for different combinations of button configurations that reflect their level of preference for each configuration. By collecting this data, the Bayesian optimizer is able to more accurately understand which configuration combinations are more popular.

1. Firstly, based on a set of determined weight values W_1 and W_2 for the two objective functions we can use an exhaustive enumeration method to find the optimal button combination for the situation.
2.
2. However, at this point the optimal Key combination is only obtained on the basis of this combination of weight values. We need to rate the obtained key combinations to the user. In this design I have defined the user's feedback as a rating mode. The rating range is 0-10, i.e., 0 is a very poor key combination, while 10 is a very satisfactory key combination for the user.
3. When the user gives a rating, the Bayesian optimizer will select new weights and perform an exhaustive search to find the best key combination for the situation and then take it to the user for

rating again. Thus we can use the user's feedback to optimize the combination of key interaction attributes.

In this way, the Bayesian optimizer iterates continuously, updating its model with each iteration based on the latest feedback from the user to find the best button configuration more accurately. The advantage of this approach is that it combines the efficiency of the algorithm with the intuitive nature of user feedback, which enables it to ensure the quality of the optimization while making effective use of the user's subjective preference information.

Experiments and results:

Based on the code in Assignment 1. First I define the objective function and the scoring system as the function multi_objective_score. its inputs are the combination of the key attributes and the objective function weights wr. where I divide the two objective function weights into wr and 1-wr. This achieves the effect that by changing the value of one of the weights one can change the value of the other, which is more convenient to use for Bayesian optimizers to perform single-parameter optimization.

```
def multi_objective_score(combination, wr):
    usage_rate = [0.1, 0.15, 0.5, 0.15, 0.2]

    Xi = [2 if comb == 0 else 0 for comb in combination[:5]]
    Yi = [1 if comb == 0 else 0 for comb in combination[5:]]
    A1 = sum([usage_rate[i] * (Xi[i] + Yi[i]) for i in range(5)])

    xi = [0 if comb == 0 else 1 for comb in combination[:5]]
    yi = [0 if comb == 0 else 2 for comb in combination[5:]]
    a2 = sum([(1-usage_rate[i]) * (xi[i] + yi[i]) for i in range(5)])

    return wr * A1 + (1-wr) * a2
```

After that, by importing itertools, I searched for all combinations of key attributes in the search space using an exhaustive search and rated them using the previously defined multi_objective_score function. Each time a combination with a better score is available, it is kept as the best combination.

```
for combination in itertools.product(range(2), repeat=10):
    score = multi_objective_score(combination, wr)
    if score > best_score:
        best_score = score
        best_combo = combination
```

And based on the code in Assignment 2, I output the result of the best combination obtained to the user and rate it, and the return value of the function is the result of the user's rating.

```
res = input("Rate this button combination (0 to 10): ")
return -float(res)
```

And in Bayesian optimizer, wr is searched from 0 to 1. This means that the optimizer will find the

best wr value between 0 and 1.

```
def run_bo(max_iter):
    bounds = [{'name': 'wr', 'type': 'continuous', 'domain': (0, 1)}]
```

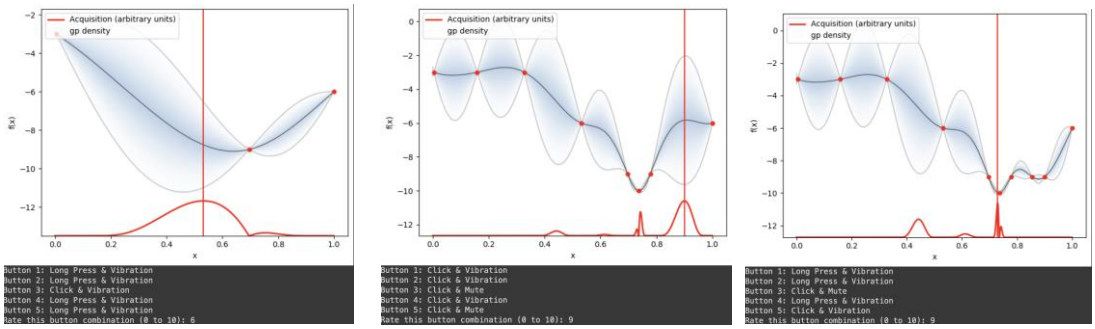
3 use scenarios:

In this task, I used each of the three use scenarios by defining three sets of frequencies about the use of the five virtual keys.

Scenario 1:

F_1	F_2	F_3	F_4	F_5
0.1	0.15	0.5	0.15	0.2

In this case, there is one key that is used significantly more than the other keys. The third and fifth buttons are used more frequently, with the third button having the highest usage rate. We guess that the third key should be in the mode of click trigger as well as mute feedback. The optimization process and results are as follows:



The final result shows that the best WR value is 0.7358143199612556. the best combination of key attributes at this value is:

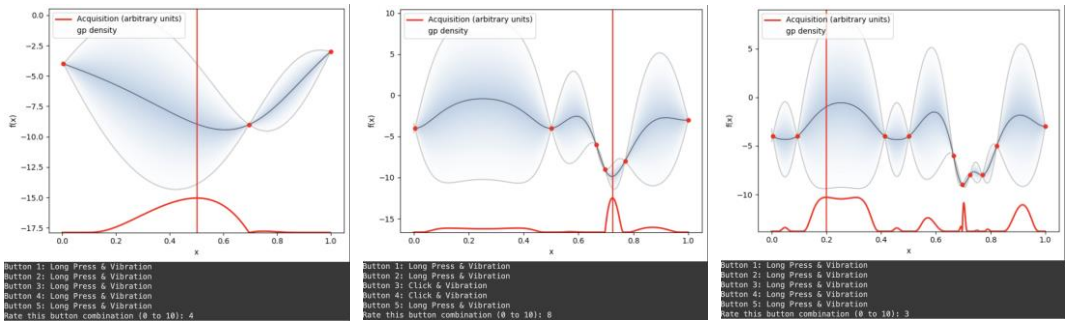
```
Button 1: Long Press & Vibration
Button 2: Long Press & Vibration
Button 3: Click & Mute
Button 4: Long Press & Vibration
Button 5: Click & Vibration
```

As you can see from the results, button 3 does use a click trigger and mute feedback mode, as well as button 5 also has a click trigger and vibration feedback mode. The other low-usage buttons are all long-press trigger and vibration feedback modes. Overall the results are consistent with the expected results.

Scenario 2:

F_1	F_2	F_3	F_4	F_5
0.15	0.1	0.3	0.3	0.15

In this case, there are two key pairs that are more heavily utilized, but there are no obvious particularly prominent highly utilized keys compared to the previous case. The third and fourth buttons are more frequently used. We guess that the third and fourth keys should be in the mode of click trigger as well as mute feedback. The optimization process and results are as follows:



The final result shows that the best WR value is 0.6964691855978616. the best combination of key attributes at this value is:

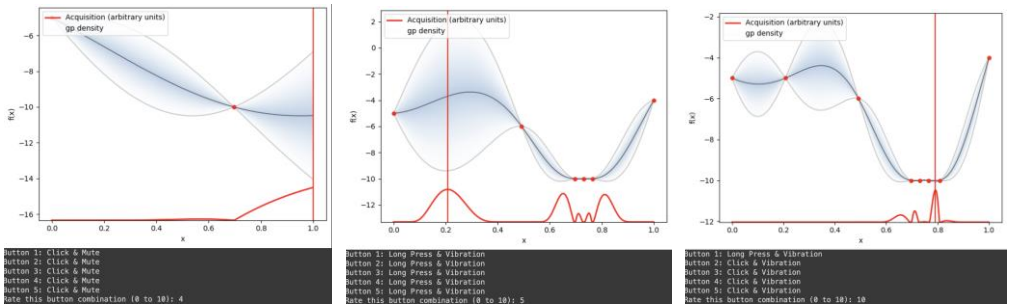
```
Button 1: Long Press & Vibration
Button 2: Long Press & Vibration
Button 3: Click & Vibration
Button 4: Click & Vibration
Button 5: Long Press & Vibration
```

From the results, we can see that button 3 and button 4 do use the click trigger mode, but with vibration feedback. The rest of the buttons use long-press trigger and vibration feedback. For ease of use, buttons 3 and 4 do have the most critical attribute of click triggering. Considering the feedback properties of the two buttons, the overall result is not exactly as expected.

Scenario 3:

F_1	F_2	F_3	F_4	F_5
0.1	0.225	0.225	0.225	0.225

In this case, there are four key pairs with high usage, and one of them has the lowest usage. We guess the first key is long-press trigger and vibration feedback, and the interaction properties of the remaining four keys should be kept consistent. The optimization process and results are as follows:



The final result shows that the best WR value is 0.8052697955828113. the best combination of key attributes at this value is:

```
Button 1: Long Press & Vibration
Button 2: Click & Vibration
Button 3: Click & Vibration
Button 4: Click & Vibration
Button 5: Click & Vibration
```

As can be seen from the results, button 1 does use the interaction attributes of long press trigger and vibration feedback. The other four buttons all adopt click trigger and vibration feedback, and all of them maintain the same interaction attributes. This indicates that these four buttons are optimized to have the same interaction attributes as expected under the scenario design with the same usage frequency.

Conclusion and Discussion:

This project implements the optimization of five virtual button interaction attributes by combining user feedback and Bayesian optimization methods in the field of user interface design for music player applications. Based on the previous task design of the combinatorial optimizer in Assignment 1, through iterative optimization and balancing the goals of ease-of-use and false-touch reduction, we developed a design flow that effectively integrates user preferences. Which not only uses the exact method of the combinatorial optimizer in A1, i.e., the exhaustive method, to achieve the global search of the design space, but also uses the Bayesian optimizer to guide the optimization process of weight assignment within the objective function through actual user ratings. The optimization results reflect that there are significant differences in user preferences for different keys and that the optimization algorithm is able to derive key configurations that are consistent with the frequency of use and expected interaction properties.

Pros:

The advantage of this project is that the user's intuitive feedback is directly introduced to ensure that the design results meet the user's actual needs, while the application of Bayesian optimization algorithm improves the efficiency and accuracy of the optimization process. In addition, for the design of the objective function, I assigned different scores according to the design objectives. This makes the optimization process more in line with the real situation.

Cons:

However, the assignment pattern of the objective function almost mostly depends on my personal subjective perception. Although in general conformity with the real situation, there is still some influence of cognitive bias. For this task, a more realistic and objective assignment system is needed to ensure the rationality of the optimization process. In addition to that, I did not find the attribute combination of “Long press & Mute” in the Bayesian optimization process. I guess it may be because the assignment system I designed is not reasonable enough that this combination cannot reach the highest score to be the best combination in the exhaustive enumeration method, so this combination never appeared in the Bayesian optimization process.