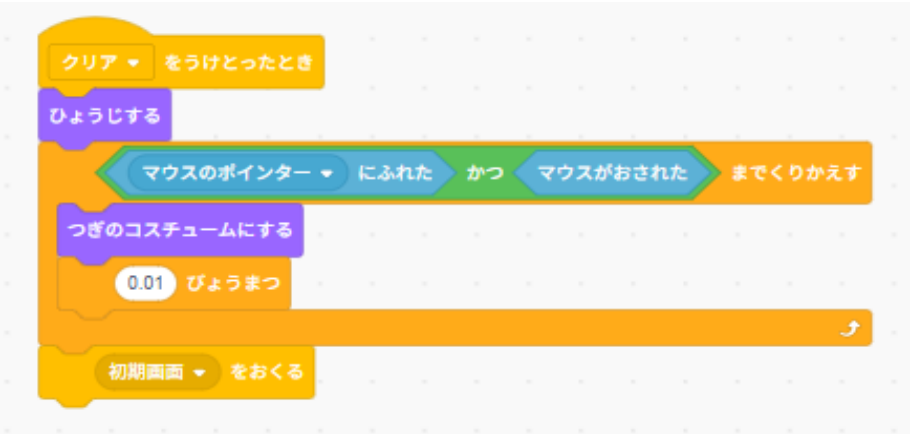
採点結果詳細【技術力】	
(1). 場合分け（例：もし）、もしくは 繰り返し（例：ずっと）、を使っている	10／10点

よくできているところ

正しく使えています。



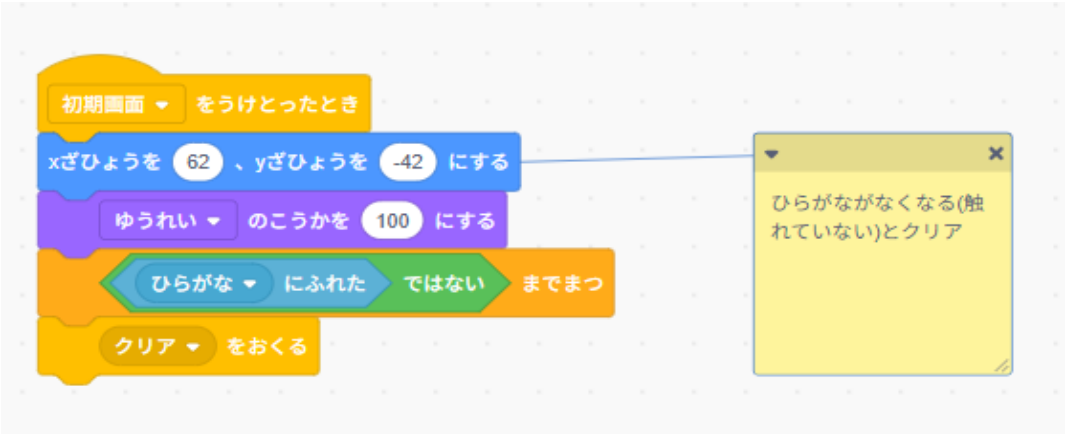
採点結果詳細【技術力】	
(1). 場合分け（例：もし）、もしくは 繰り返し（例：ずっと）、を使っている	
改善できそうなところ	特になし

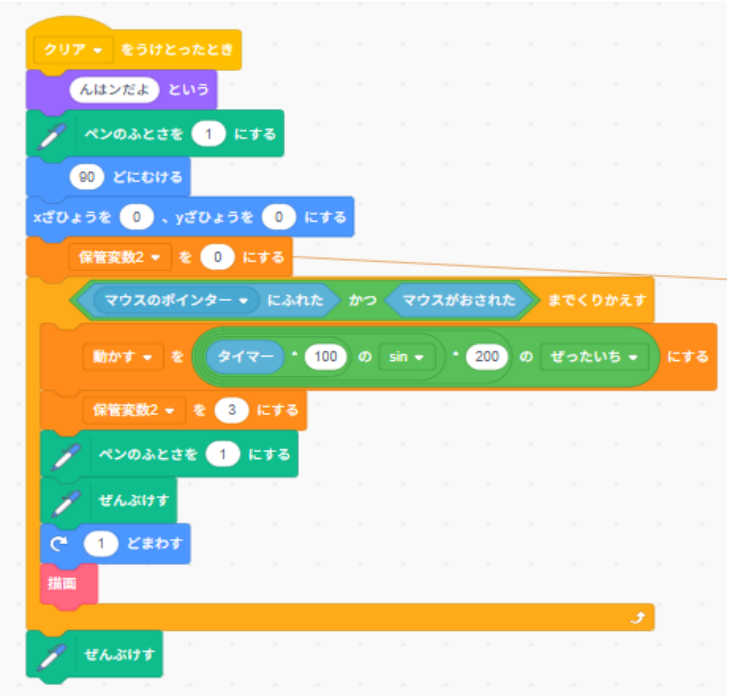
採点結果詳細【技術力】	
(1). 場合分け（例：もし）、もしくは 繰り返し（例：ずっと）、を使っている	
改善案	特になし

採点結果詳細【技術力】	
(2). メッセージを使っている	10／10点

よくできているところ

正しく使えています。





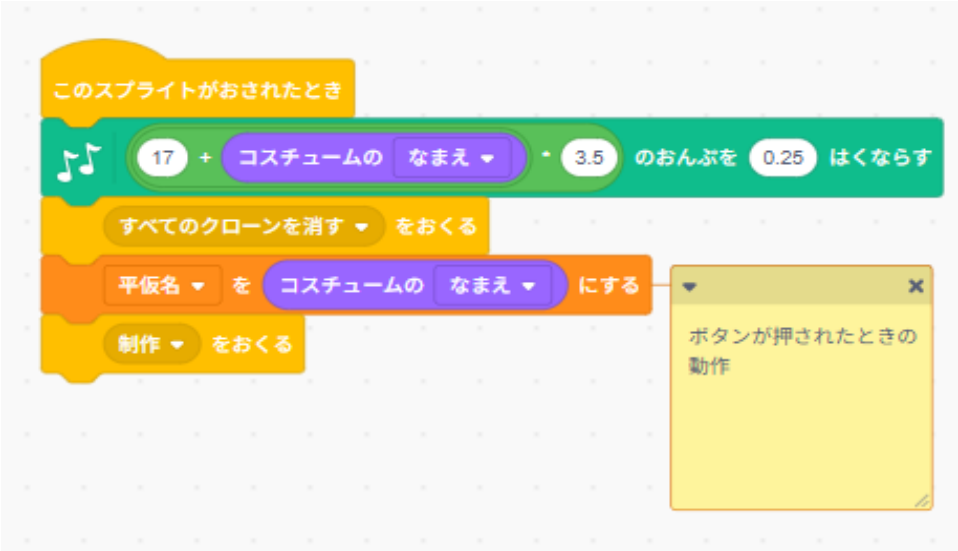
採点結果詳細【技術力】	
(2). メッセージを使っている	
改善できそうなところ	特になし

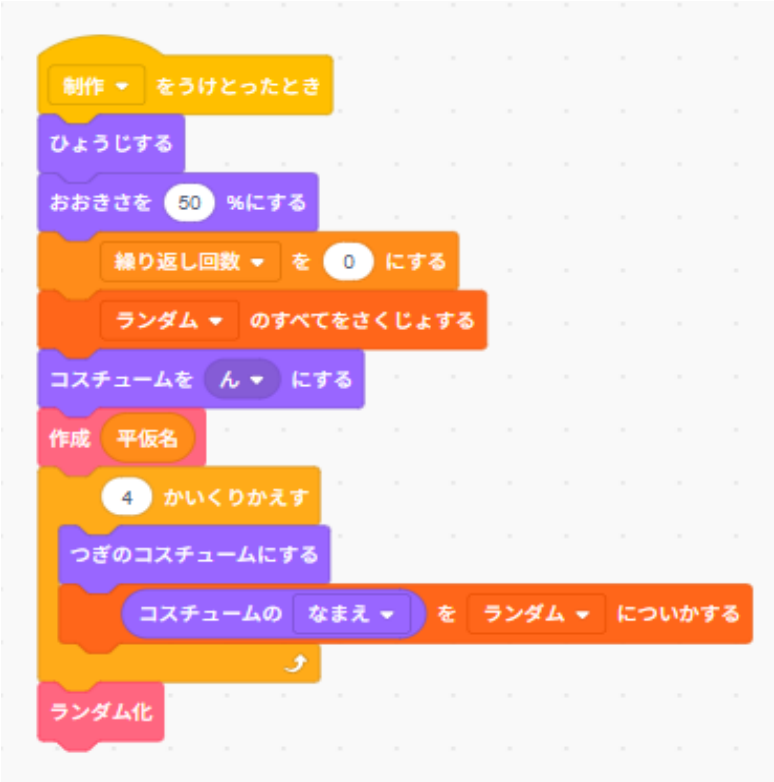
採点結果詳細【技術力】	
(2). メッセージを使っている	
改善案	特になし

採点結果詳細【技術力】	
(3). 変数を使っている	10／10点

よくできているところ

正しく使えています。





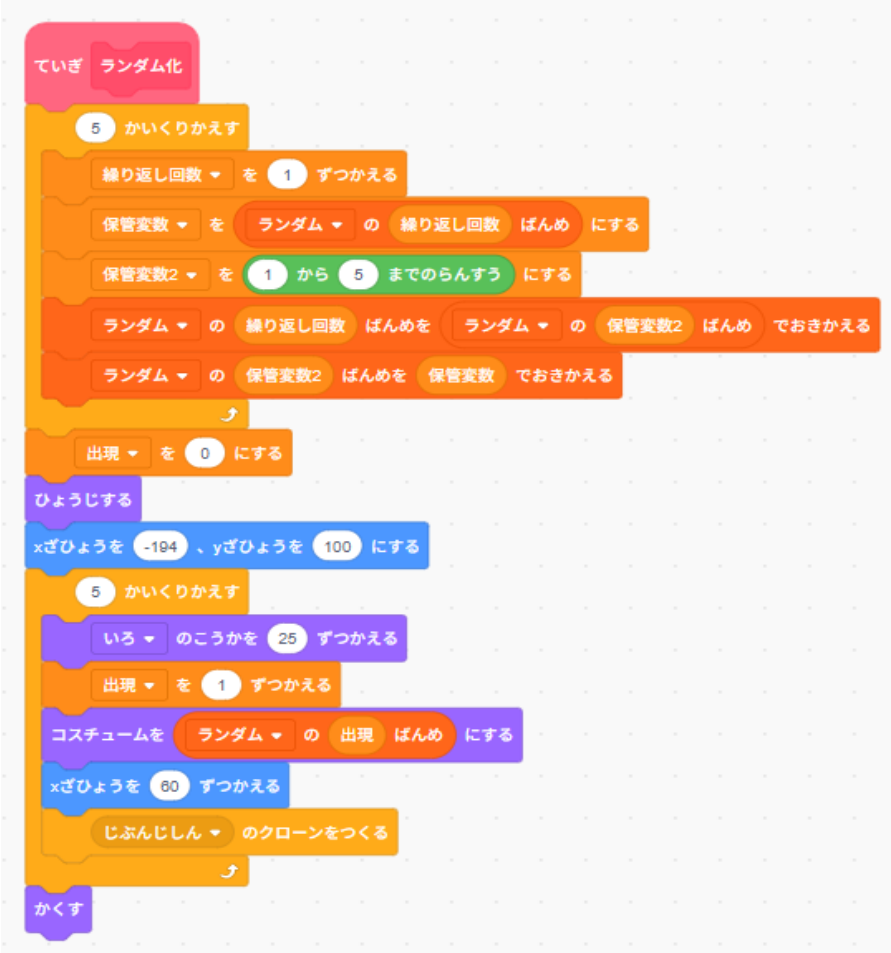
採点結果詳細【技術力】	
(3). 変数を使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(3). 変数を使っている	
改善案	特になし

採点結果詳細【技術力】	
(4). クローンを使っている	10／10点

よくできているところ

正しく使えています。



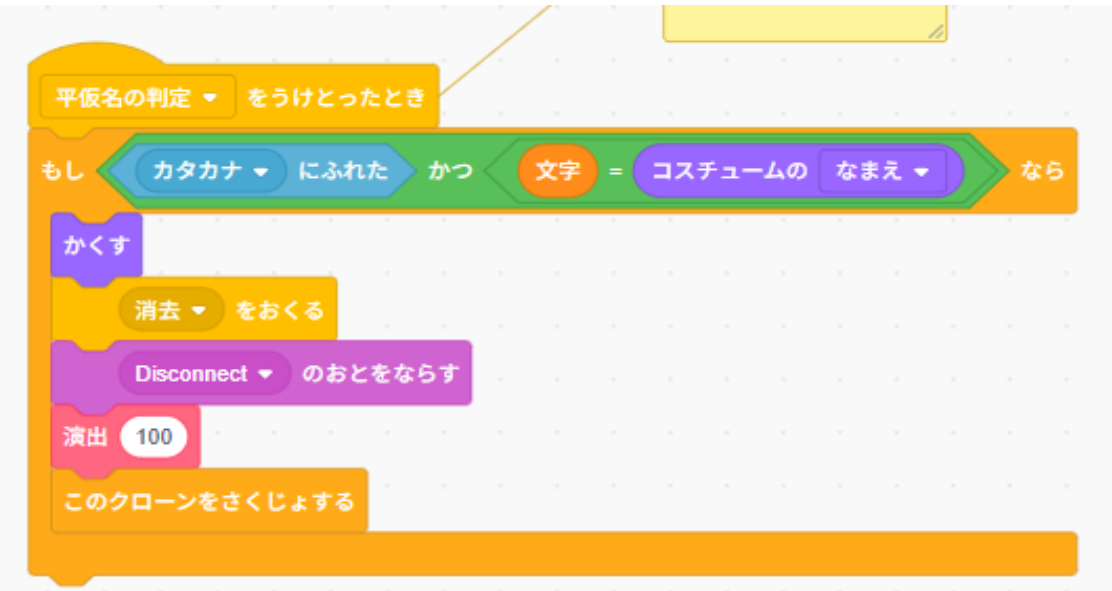
採点結果詳細【技術力】	
(4). クローンを使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(4). クローンを使っている	
改善案	特になし

採点結果詳細【技術力】	
(5). ブロック定義を使っている	10／10点

よくできているところ

正しく使えています。



採点結果詳細【技術力】	
(5). ブロック定義を使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(5). ブロック定義を使っている	
改善案	特になし

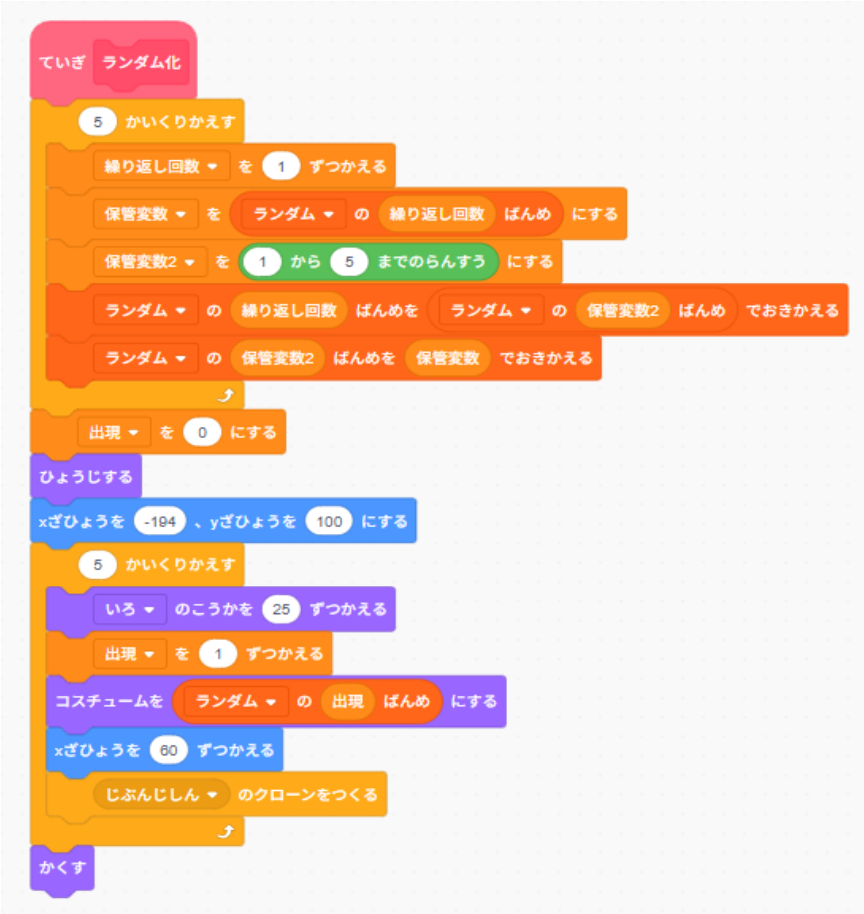
採点結果詳細【技術力】

(6). リストを使っている

10／10点

よくできているところ

正しく使えています。



採点結果詳細【技術力】	
(6). リストを使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(6). リストを使っている	
改善案	特になし

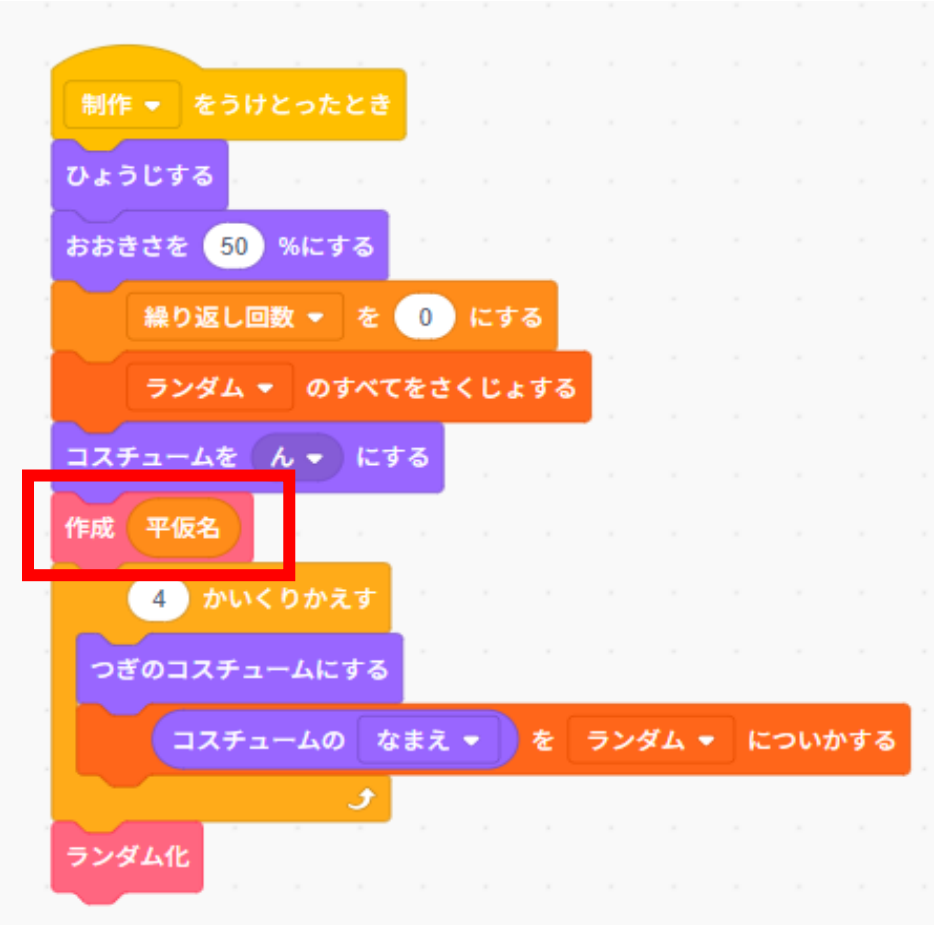
採点結果詳細【技術力】	
(7). プログラムの品質 （１つの指摘につき３点減点）	10／10点

よくできているところ	
	とても読みやすいプログラムです。 修正したほうがよいところは見受けられませんでした。

採点結果詳細【技術力】
(7). プログラムの品質 （１つの指摘につき３点減点）

改善できそうなところ

減点ではありませんが、気になったところは以下のプログラムです。



採点結果詳細【技術力】
(7). プログラムの品質 （１つの指摘につき３点減点）

改善案	
	メニューから遊びたい文字を選択しますが、 その都度「あ～お」のリストを作っていると思います。 「あいうえお」「かきくけこ」・・・のリストは 決まっている内容なので最初に一度だけ作成しておく、という方法があります。 今回は最大５文字しか入らないリストを作っているだけなので パフォーマンス（速度）に影響はないと思いますが、 大きなサイズのリストを作る場合は、作成にコスト（時間・メモリ圧迫）が かかります。 そういった場面に遭遇したら、上記の方法でプログラミングしてほしいと思います。