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。

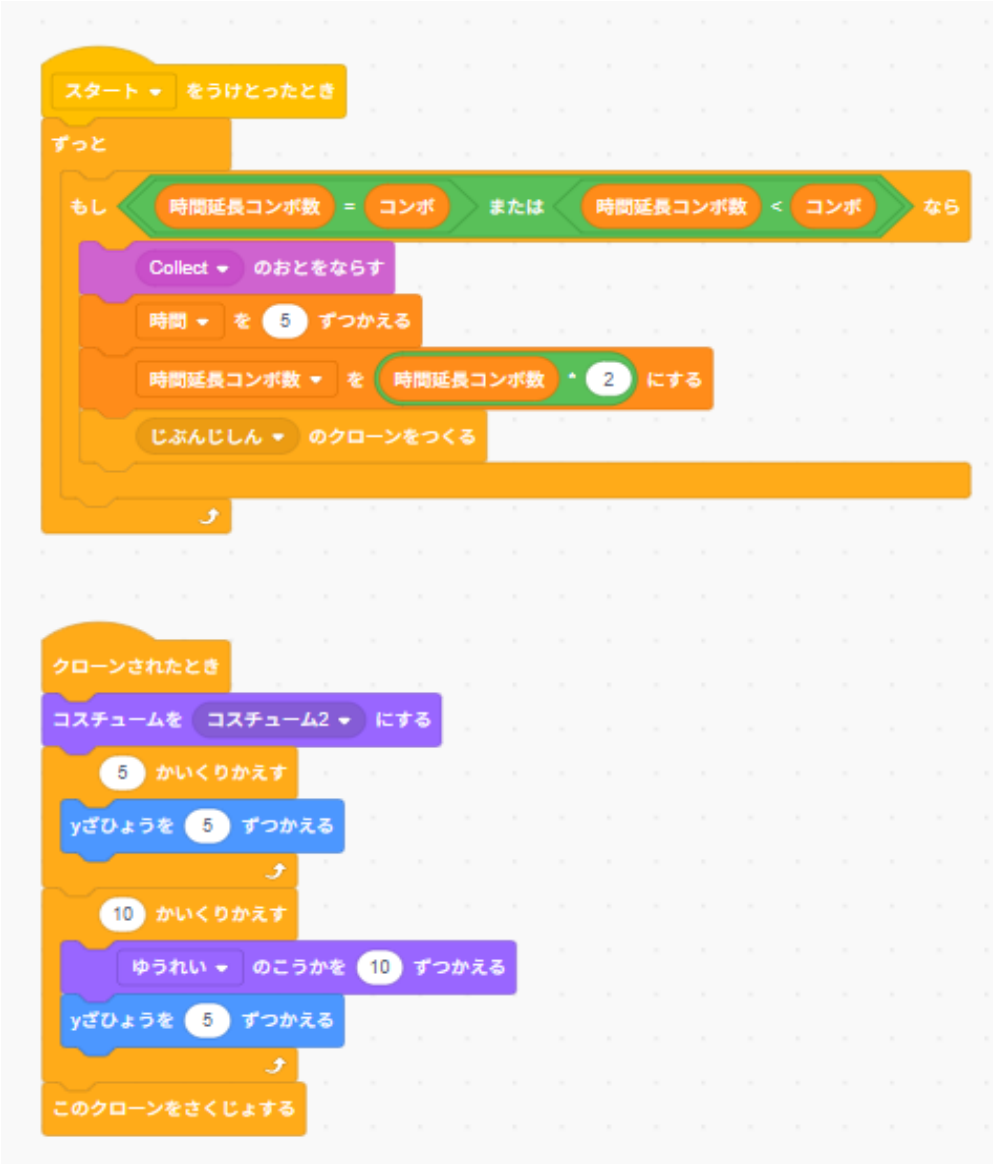
採点結果詳細【技術力】	
(3). 変数を使っている	
改善できそうなところ	特になし

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	0 ／10点

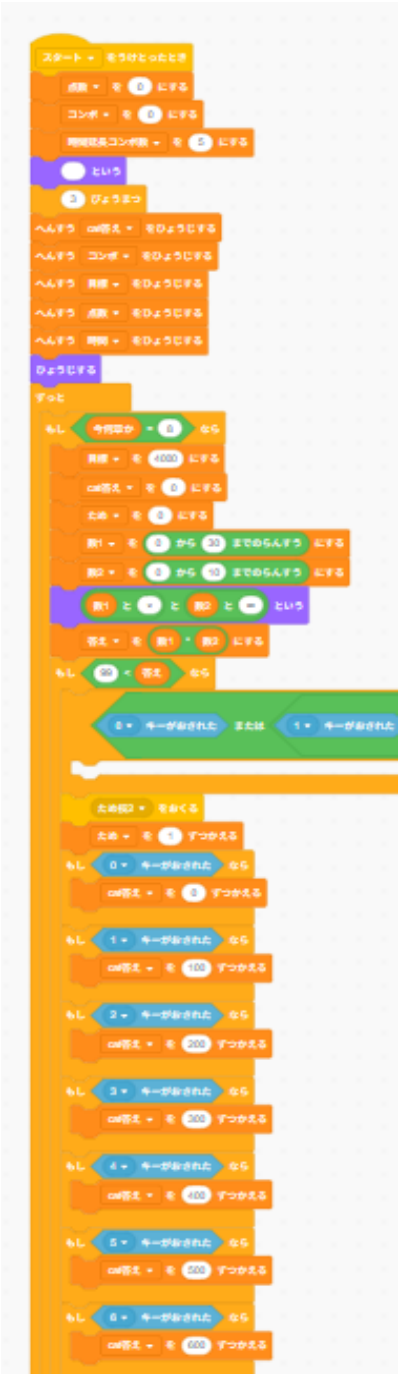
よくできているところ

この作品ではブロック定義は使われていません。

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

下記のような長いプログラムは、ブロック定義を使って分割しましょう。



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	0 ／10点
よくできているところ	この作品では使われていないようです。

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）	0 ／10点

よくできているところ	
	改善できるところがあります。

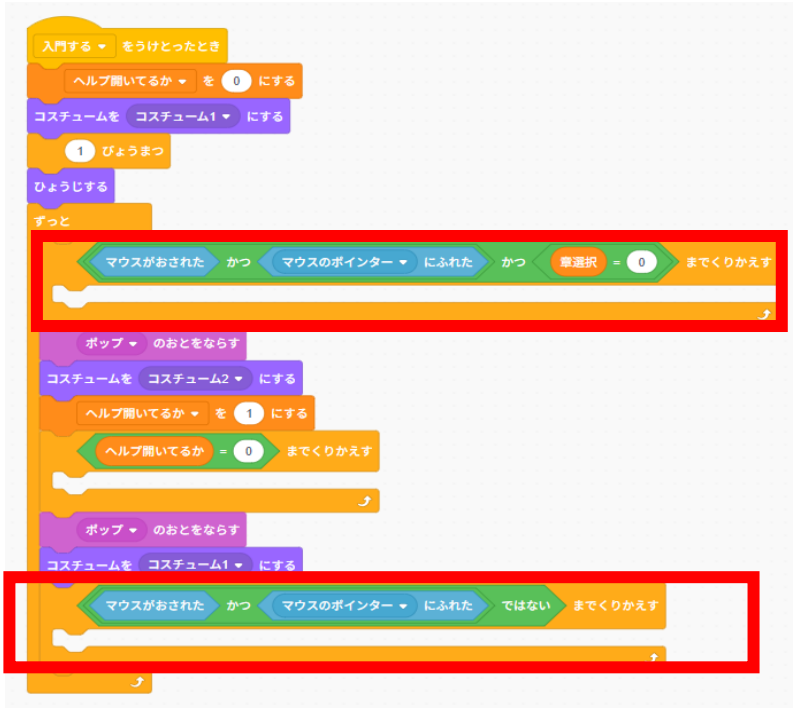
採点結果詳細【技術力】

(7). プログラムの品質
（１つの指摘につき３点減点）

改善できそうなところ

分かりにくいスプライト名が散見されます。

「●までくりかえす」ブロックが使われていますが
繰り返す処理が書かれていないものがあります。



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

スプライト名、変数名やメッセージ名は分かりやすい名前にしましょう。

ただ待つだけであれば「●までまつ」というブロックを使いましょう。