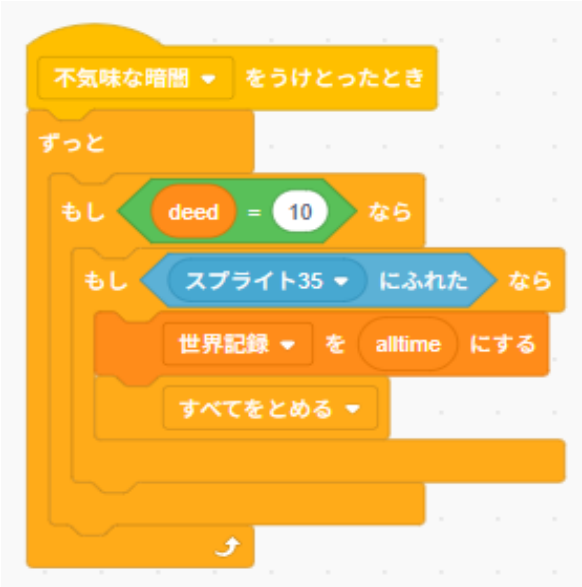
採点結果詳細【技術力】	
(1). 場合分け（例：もし）、もしくは 繰り返し（例：ずっと）、を使っている	10／10点

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	0 ／10点

よくできているところ

この作品でクローンは使われていないようです。

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	同じような処理（特徴）をもつスプライトが多数あります。

採点結果詳細【技術力】	
(4). クローンを使っている	
改善案	そのようなスプライトはクローンを使ってひとつにまとめることができます。

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ないようです。

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

プログラムの処理が長い所や、
同じ処理が複数存在するときに、ブロック定義が役に立ちます。

例えば、下記の長いプログラムをブロック定義を使って分割します。

採点結果詳細【技術力】
(5). ブロック定義を使っている

改善案	下記のように分割することができます。 長すぎるプログラムは読みにくいのでこのように改善できます。

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

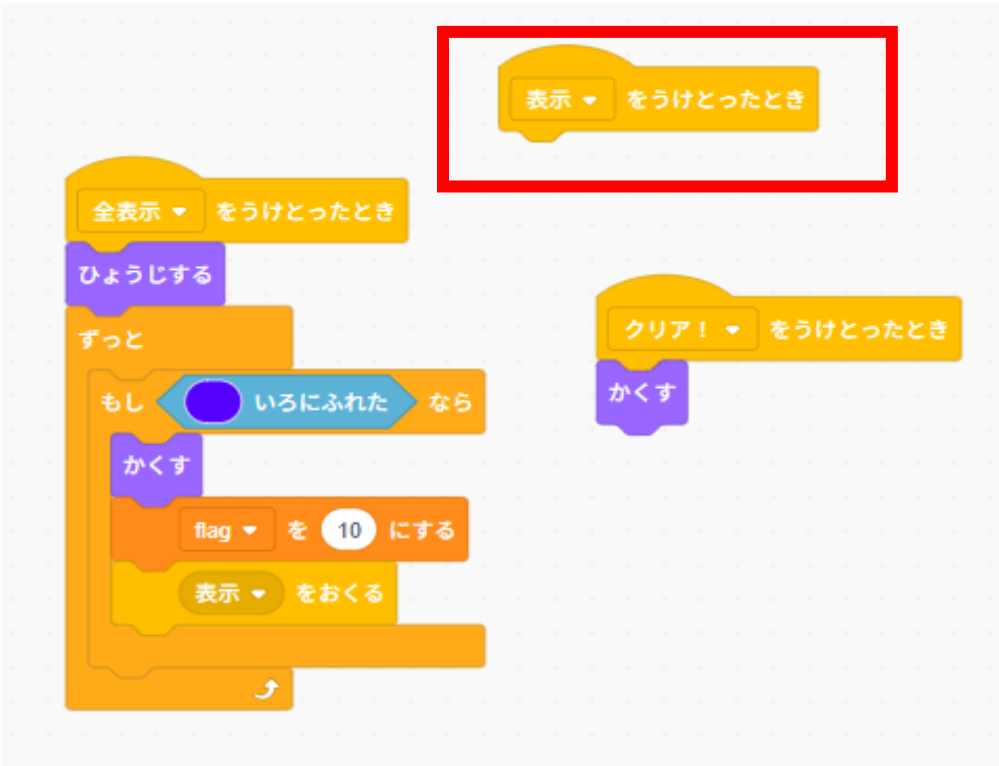
改善できるところがあります。

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使われています。
・スプライト１，２，・・・

使われていないブロックがあります。



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。 使わないブロックは削除しましょう。