

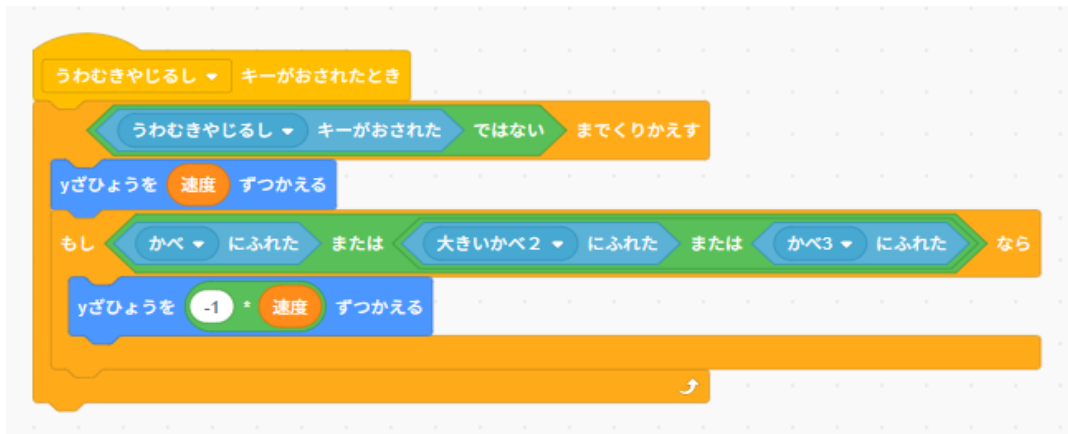
採点結果詳細【技術力】

(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

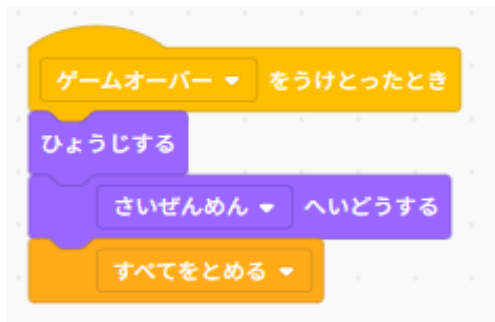
採点結果詳細【技術力】

(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

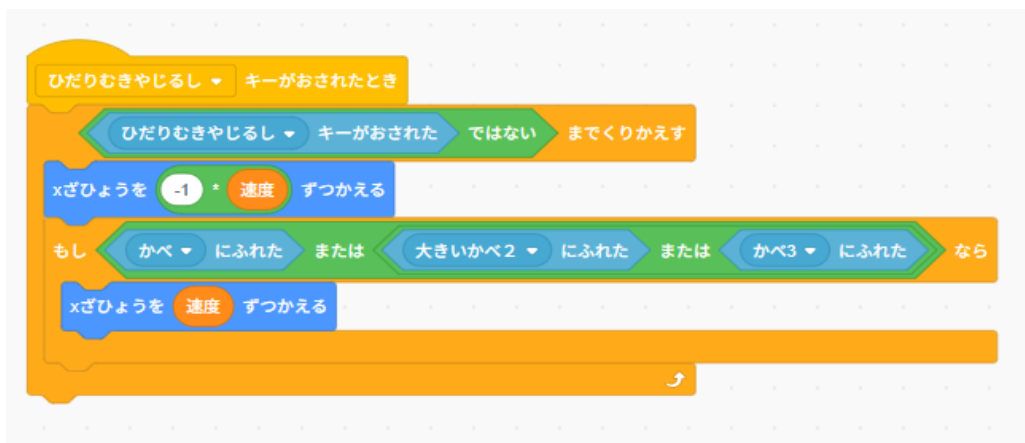
採点結果詳細【技術力】

(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

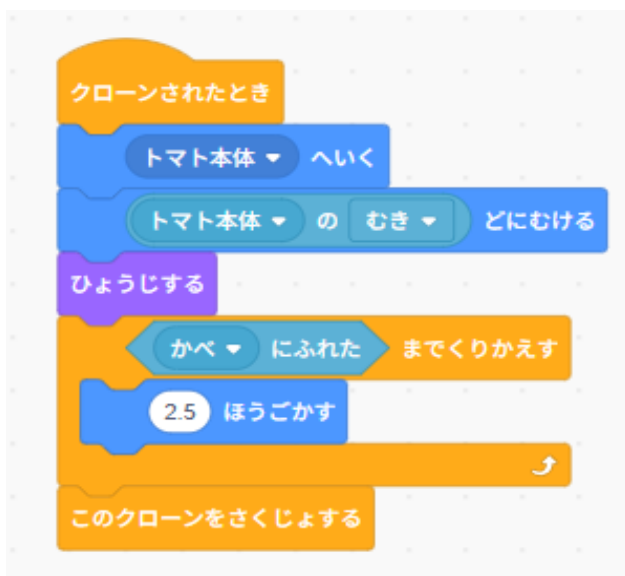
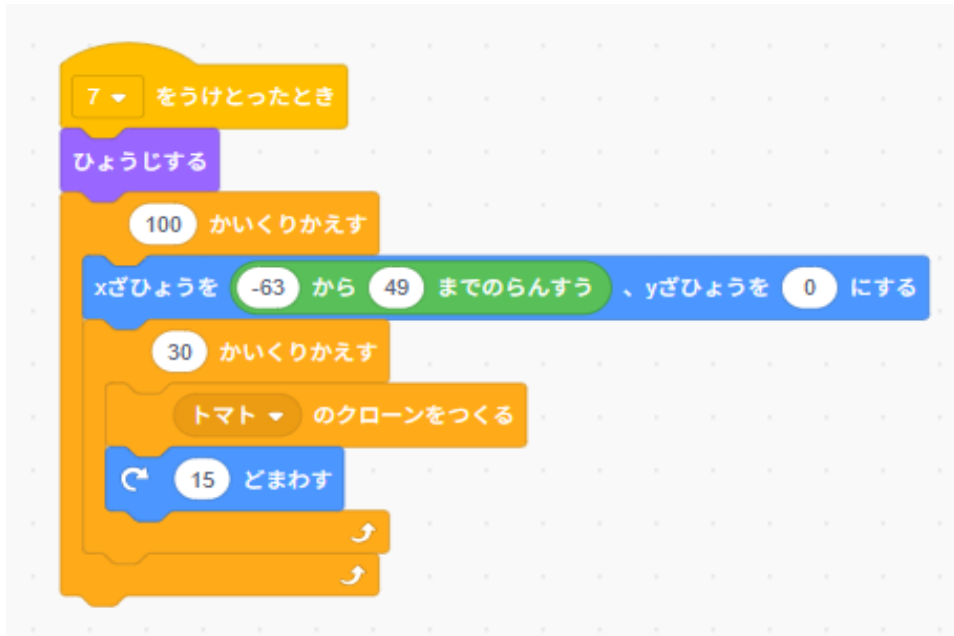
採点結果詳細【技術力】

(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

ブロック定義を使うと、長いプログラムを分割することができます。
また、同じようなプログラムを一つにまとめることができます。
プログラムが読みやすくなるのでぜひ使ってみてください。

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

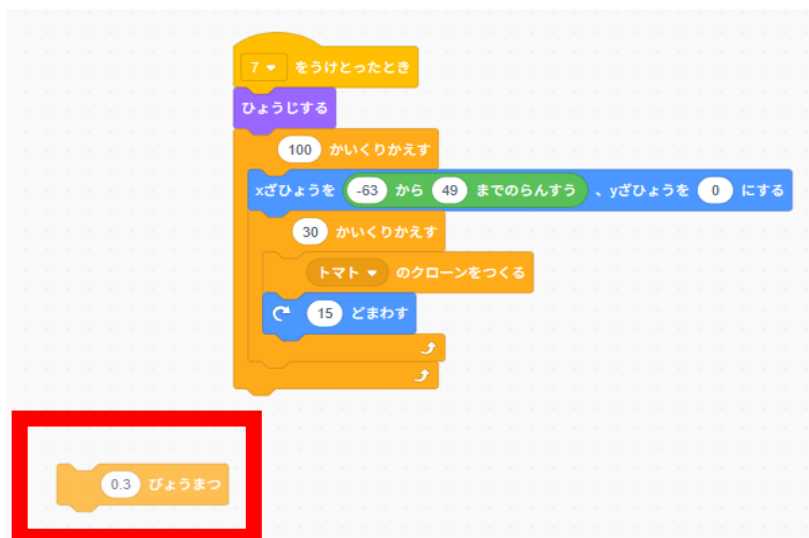
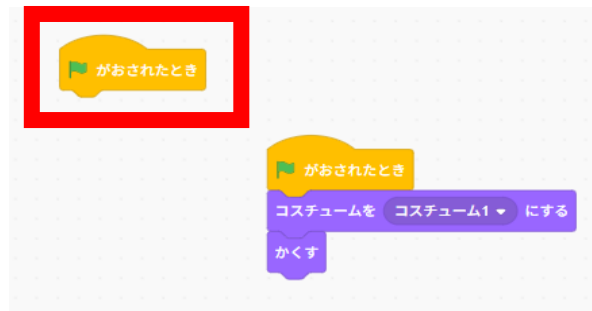
採点結果詳細【技術力】		
(7). プログラムの品質 (1つの指摘につき3点減点)		0 / 10点
よくできているところ	プログラムの品質は全体的に高いと言えます。	

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

使われていないブロックが散見されます。



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

使わないブロックは削除しましょう。

