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。

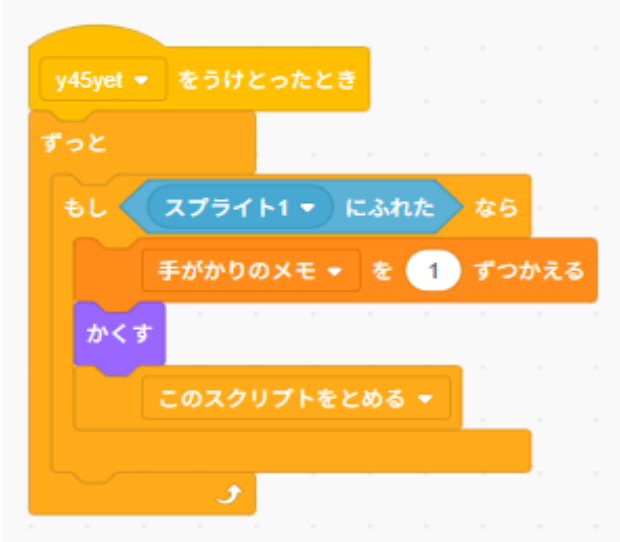
採点結果詳細【技術力】	
(2). メッセージを使っている	
改善できそうなところ	特になし

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

この作品ではクローンは使われていません。

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	スプライト「Ball、Ball2、Ball3、Ball4」はよく似たプログラムです。 これらはクローンを使うことで一つのスプライトにまとめることができます。 「スプライト18」と「スプライト19」も同様です。

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

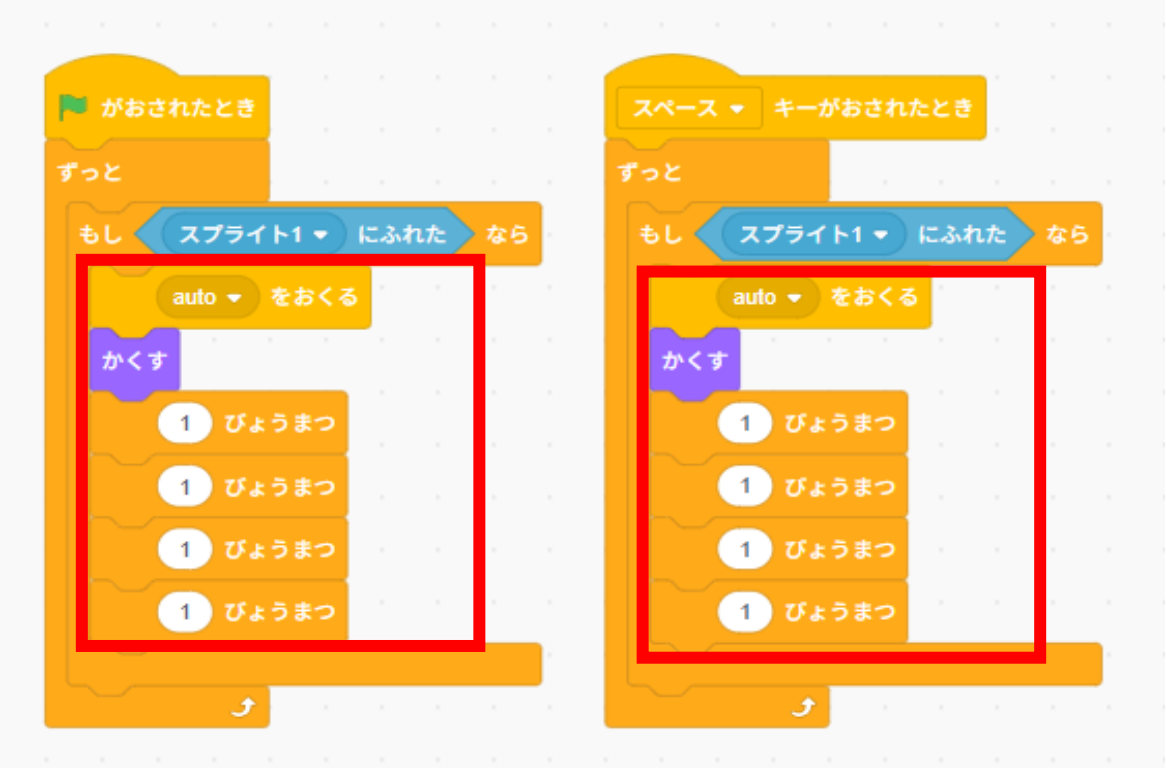
よくできているところ

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

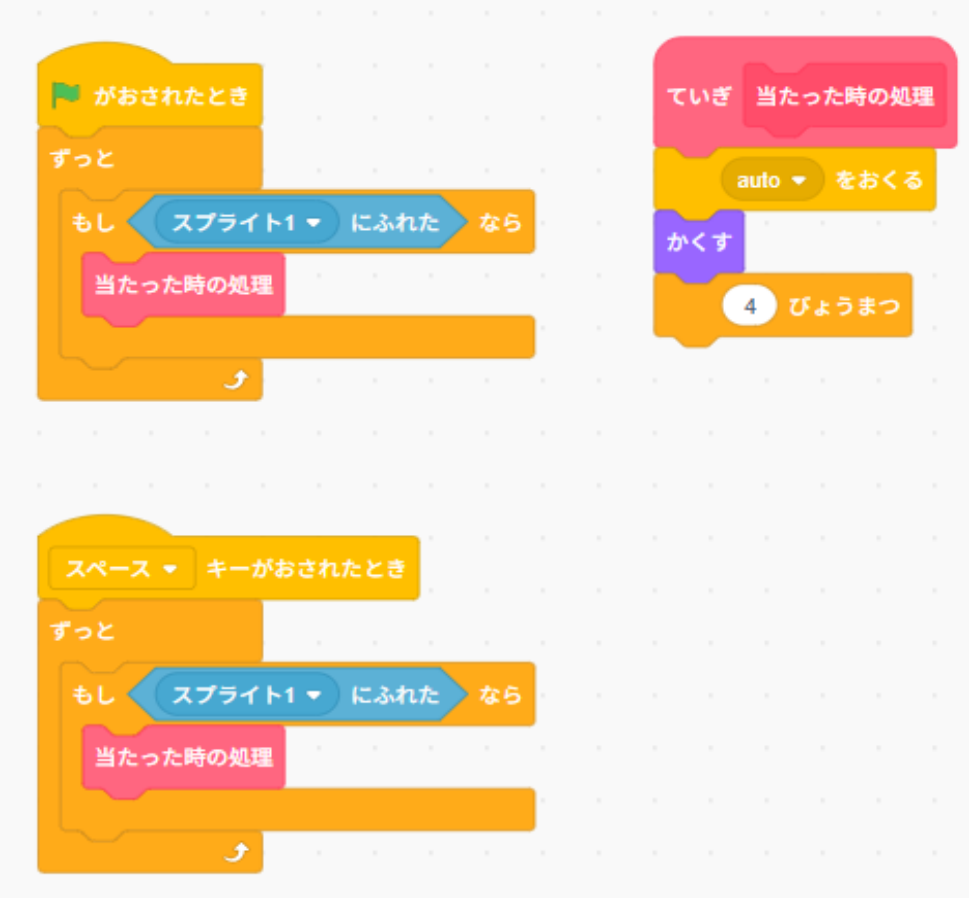
下記はスプライト「Ball」のプログラムです。
赤枠のプログラムが同じです。
このようなプログラムはブロック定義を使ってまとめることができます。



採点結果詳細【技術力】
(5). ブロック定義を使っている

改善案

ブロック定義を使えば、
同じプログラムを何度も書かなくて済みます。



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

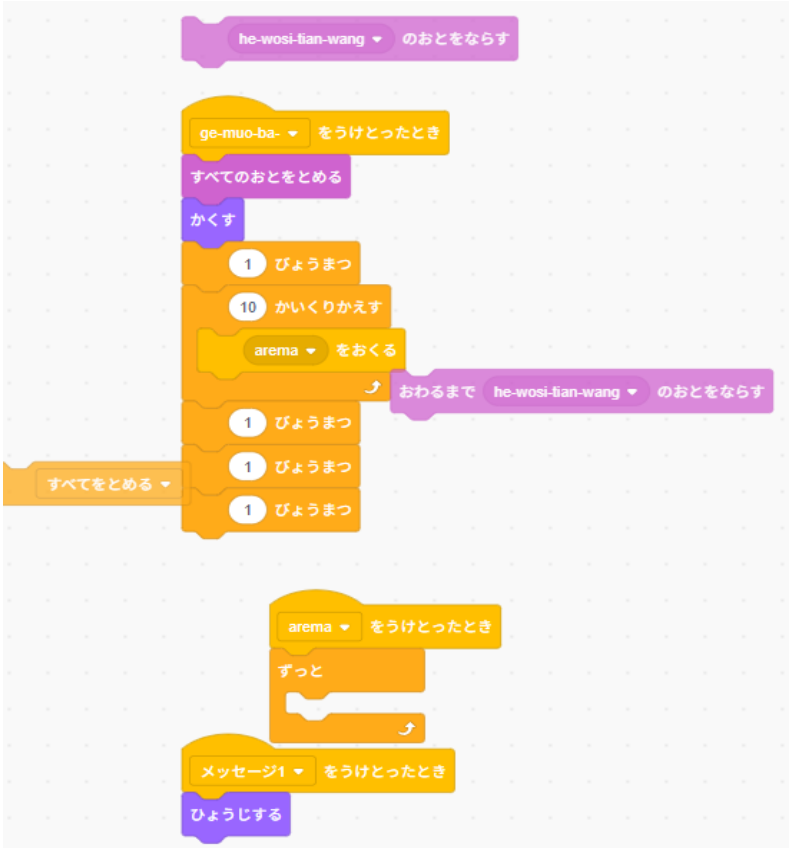
改善できるところがあります。

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。

使われないプログラムが散見されます。



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	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。 使わないプログラムは削除しましょう。