パックマンもどき ~迷路ありバージョン~ レベル1

今回のキモ

グリッドに沿って迷路を作り、キャラクターの動きもグリッド上に限定します。

グリッドに沿って迷路を作る 1

ステージに迷路を描く際、グリッドに沿って作りたいと思います。 ① 背景を選ぶ場面の一覧から、「Xy-grid-20px」を選んでください。



ドリームプラス https://drmpls.com

② 背景にてグリッドの上に迷路を描いていきましょう。
 [直線ツール]で、太さ12(くらい)の直線を描いていきます。
 色も任意ですが、完成例では下図のような設定にしました。

✓ ▼ 枠線 □ ▼ 12	0
色 48	
鮮やかさ 100	
	j
明るさ 100	

③ Shift キーを押しながらドラッグすると、直線を水平・垂直に引くことができます。



③ まずは、下図のように大外の枠とオバケのホーム部分を作ってください。 壁の線はできる限りグリッド線に合わせてください。 位置の微調整は矢印キーでできます。



あとは任意に迷路を描いてくださって構いませんが、四隅にパワーフードを置き ますのでその点は考慮してください。

プレイヤーやオバケが通る道の部分は2マス分(40px)にしてください。 完成例は下図のようにしました。



ドリームプラス https://drmpls.com

4 [選択] ツールにしてグリッドを選択してください。



⑤ その状態で [ゴミ箱] アイコンをクリックしてください。 グリッド線が消えます。



パックマンもどき~迷路あり レベル1 https://scratch.mit.edu/projects/708192946

⑥ 黒い背景を作ります。[四角形] ツールを選択し、背景全体を黒い四角形で塗り つぶしてください。



⑦ 四角を書いた直後はそれが選択状態になっているので、そのまま [最背面] アイ コンをクリックしてください。



ドリームプラス https://drmpls.com

5

⑧ これで、黒い背景に迷路がある状態になりました。



2 プレイヤーを作る

2-1 スプライトを描く

プレイヤーとなるスプライトを作ってください。 大きさは迷路の幅に収まるようにしてください。大きさの欄で調整しましょう。

コスチュームは2つ、一つは円。もうひとつはそれを複製してロが開いた状態のものとしてください。

※円の中心をスプライトの中心に合わせることを忘れないでください。



2-2 プログラミング

以下、各段階で動作確認しながら進めて行ってください。

【問題2-1】

- (1)緑の旗が押された時、
 - 位置を x=0, y=-40
 - 向きを-90 度
 - 表示する
 - としてください。
- (2) 左矢印キーが押されたら
 - -90 度に向ける
 - 20 歩動かす
 - としてください。

動きが早いので、0.1秒待つようにしてください。



【問題2-1】

(3) 左矢印キーが押されているとき、コスチュームが次々に切り替わるようにして ください。

(プレイヤーが口をパクパクさせているような動き)

(4) 右、上、下にもそれぞれの矢印キーで同様に動くようにしてください。

【問題2-1】解答例 (3)(4)

**** 0.1 秒待つ もし 左向き矢印 ・ キーが押された なら -90 度に向ける 20 歩動かす 次のコスチュームにする ・0 度に向ける 20 歩動かす 次のコスチュームにする ・0 度に向ける 20 歩動かす 次のコスチュームにする ・0 度に向ける 20 歩動かす 次のコスチュームにする ・0 度に向ける 20 歩動かす 次のコスチュームにする	表示する								
0.1 秒待つ もし 左向き矢印 ・ キーが押された なら -90 度に向ける 20 歩動かす 次のコスチュームにする もし 上向き矢印 ・ キーが押された なら 90 度に向ける 20 歩動かす 次のコスチュームにする									
0.1 砂符つ もし 左向き矢印・キーが押された なら -90 度に向ける 第 20 歩動かす 第 次のコスチュームにする 5 90 度に向ける 5 20 歩動かす 5 90 度に向ける 5 20 歩動かす 5 90 度に向ける 5 20 歩動かす 5	r້ວと ∼								
 5 し 左向き矢印 → キーが押された なら 90 度に向ける 20 歩動かす 次のコスチュームにする 5 し 右向き矢印 → キーが押された なら 90 度に向ける 20 歩動かす 、 20 歩動かす 、 30 度に向ける 30 度に向ける 30 度に向ける 30 度に向ける 30 度に向ける 30 変動かす 、 30 度に向ける 30 度に向ける 30 度に向ける 30 度に向ける 30 変動かす 、 30 なら 、 30 度に向ける 30 方 、 31 方 、 31 方 、 32 歩動かす 、 33 方 、 34 方 、 35 方 、 35 方 、 36 方 、 37 方 、 37 方 、 38 方 、 39 方 、 39 方 、 30 方 、 30 方 、 31 方 、 31 方 、 31 方 、 32 方 、 34 方 、 35 方 、 35 方 、 36 方 、 37 方 、 38 方 、 39 方 、 39 方 、 30 方 、 31 方 、 32 方 、 31 方 、 <		0.1 秒待つ	, . 						
-90 度に向ける 20 歩動かす 次のコスチュームにする 40 度に向ける 90 度に向ける 20 歩動かす 次のコスチュームにする 10 度に向ける 20 歩動かす 次のコスチュームにする 10 度に向ける 20 歩動かす 次のコスチュームにする 180 度に向ける 20 歩動かす 次のコスチュームにする	もしく	左向き矢印	+	ーが押	され	2	なら		
20 歩動かす 次のコスチュームにする もし 右向き矢印・キーが押された なら 90 度に向ける 20 歩動かす 次のコスチュームにする もし 上向き矢印・キーが押された なら 0 度に向ける 20 歩動かす 次のコスチュームにする もし 下向き矢印・キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする		-90 度に「	向ける						
次のコスチュームにする もし 石向き矢印・キーが押された なら 90 度に向ける 第二、第二、第二、第二、第二、第二、第二、第二、第二、第二、第二、第二、第二、第	~_	20 歩動か	्व						
まし 右向き矢印・キーが押された なら 90 度に向ける 20 歩動かす 次のコスチュームにする 3 ちし 上向き矢印・キーが押された なら 0 度に向ける 20 歩動かす 次のコスチュームにする 3 ちし 下向き矢印・キーが押された なら 180 度に向ける 次のコスチュームにする 3	Nh a								
 もし 白向き矢印・キーが押された、なら 90 度に向ける 40 参動かす 40 参動かす 40 た向ける 40 度に向ける 40 かす 40 たのける 40 度に向ける 40 かす 40 かす 40 かす 40 たのける 40 かす 40 かす 41 かが押された 40 たのける 41 かかす 41 かかかす 41 かかす 41 かかす	RO		ट 7 ବ						
 30 名向き矢印 ◆ キーが押された なら 90 度に向ける 20 歩動かす 次のコスチュームにする 30 度に向ける 20 歩動かす 次のコスチュームにする 30 度に向ける 30 歩動かす 30 歩動かす 30 次のコスチュームにする 	\sim	(++++		1.24m	و مار بادر		* *		
90 度に向ける 20 歩動かす 次のコスチュームにする もし 上向き矢印 → キーが押された なら 0 度に向ける 20 歩動かす 次のコスチュームにする もし 下向き矢印 → キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする	5U (石向ざ矢印			ani	<u>ح</u>	<i></i> кь		
20 歩動かす 次のコスチュームにする もし 上向き矢印 → キーが押された なら 0 度に向ける 20 歩動かす 次のコスチュームにする もし 下向き矢印 → キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする		90 度に向	ける						
次のコスチュームにする もし 上向き矢印 ▼ キーが押された なら ① 度に向ける ② 歩動かす 次のコスチュームにする もし 下向き矢印 ▼ キーが押された なら 180 度に向ける ② 歩動かす 次のコスチュームにする		20 歩動か	<u>क</u>						
 もし 上向き矢印 → キーが押された なら ① 度に向ける 20 歩動かす 次のコスチュームにする もし 下向き矢印 → キーが押された なら 180 度に向ける 20 歩助かす 次のコスチュームにする 	次の	コスチューム	こする						
 bl 上向き矢印 ◆ キーが押された なら ① 度に向ける ② 歩動かす 次のコスチュームにする bl 下向き矢印 ◆ キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする 									
 0 度に向ける 20 歩動かす 次のコスチュームにする 50 下向き矢印 ▼ キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする 	もしく	上向き矢印	+	ーが押	され	<u>ب</u>	なら		
20 歩動かす 次のコスチュームにする もし 下向き矢印 → キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする		0 度に向	ける						
次のコスチュームにする ちし 下向き矢印 → キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする	~ _	20 歩動か	ੇ ਹੋ						
xのコステュームにする もし 下向き矢印 ● キーが押された なら 180 度に向ける 20 歩動かす 次のコスチュームにする	Nh a								
もし 下向き矢印 → キーが押された 180 度に向ける 20 歩動かす 次のコスチュームにする	жo		C98						
FPI3矢は ◆ キーが押された なち 180 度に向ける 20 歩動かす 次のコスチュームにする		(TH *F		1.2400	at da d				
180 度に向ける 20 歩動かす 次のコスチュームにする		ГЮЗХЦ			anı	۲	<i></i> кь		
20 歩動かす 次のコスチュームにする		180 度にI	向ける						
次のコスチュームにする		20 歩動か	ক						
	次の	コスチューム	にする						
3									
		٦							

この時点では、壁をすり抜けます。

そこで、壁の色に触れたら、進む値を打ち消す動きを追加してください。



壁をすり抜けなくなれば OK です。

【問題2-1】

(5) A 地点から B 地点へワープするように、また逆に B 地点から A 地点へワープするようにしてください。



(ヒント)

左に動いている時に A 地点に来たら B 地点へワープ 右に動いている時に B 地点に来たら A 地点へワープ

x 座標で判定しましょう。x<-230、230<x とするのがおすすめです。

(試作ウラ話)

ワープ先で動かなくなってしまう現象が見られました。

その時、ワープ先の座標を230とか-230としていました。

これは、プレイヤーの動きを20ずつ動かすようにしていることと壁に当たった時 にそれを打ち消す-20ずつ動かすようにしていることと関係しています。

ですので、そのような不具合を防止するためにはワープ先の座標は 20 の倍数が良いです。(220 とか 240 ということ)

ここまでで、プレイヤーが迷路内で自由に動くようになりました。

【問題2-1】解答例

(5)	
- e -	もし 左向き矢印 - キーが押された なら	
	-90 度に向ける -91	
	20 歩動かす	
	次のコスチュームにする	
	もし x座標 < -230 なら	追加
	x座標を 240 にする	
	もし 右向き矢印 → キーが押された なら	
	90 度に向ける	
	20 歩動かす	
	次のコスチュームにする	
	もし 230 < x座標 なら い いいいいい	追加
-	x座標を -240 にする	
	at the form at stimulate and	

3 パワーフード作成

3-1 スプライトを描く

ほぼプレイヤーと同じ大きさの塗りつぶし円を描きましょう。 中心に合わせることを忘れずに。



3-2 プログラミング

【問題3-2】

下図のように4つ表示するプログラムを作りましょう。 クローンで作ってください。



わかりやすいように、グリッドも表示して示すと次の図のようになります。



- 緑の旗が押された時、隠してください。
- 「パワーフード生成」というメッセージを作り、それを受け取ったらクローンす るようにしてください。
- クローンされたら表示してください。

ここまで作ったら、「パワーフード生成」のブロックをクリックして動作確認しま しょう。

うまく動いたら、ステージのコードで緑の旗が押された時に「パワーフード生成」 を送るようにしてください。

緑の旗をクリックして動作確認しましょう。

【問題3-2】完成例

パワーフード



ステージ



4 ドット作成

4-1 生成タイミング

パワーフード生成の後のタイミングでドット作成するために、パワーフードが4 つ目のクローンを作成した後に「ドット生成を送る」を追加してください。



4-2 ドットのスプライトを描き、クローンで並べる

①パワーフードとほぼ同様、黄色い塗りつぶし円を描いてください。描くときは大きくて構いません。中心を合わせることを忘れずに。
 表示する大きさは小さくしてください。

②左上のパワーフードと同じ座標を開始点として、右下のパワーフードの座標まで 20px ずつの間隔でクローンしましょう。

下図のようになります。



x 方向に繰り返す回数、y 方向に繰り返す回数など考えながら作ってみましょう。 一時的にパワーフードや壁と重なりますが、後で削除処理を加えます。 完成例は次のページです。

15





4-3 余計なドットの削除



ドリームプラス https://drmpls.com

ですので、クローンしたタイミングで、その条件の時にクローンを削除しましょう。

- A), C)についてはそれぞれの「スプライトに触れた」で判定可能
- B),「色に触れた」で判定可能。
- D)についてのみ、座標で判定しましょう。



うまくできれば、下図のようになります。



完成例は次のページ。

パックマンもどき〜迷路あり レベル1

4-3 不要なドットを削除する完成例

クローンされたとき 表示する もし パワーフード • に触れた または プレイヤー • に触れた または 色に触れた なら このクローンを削除する	A) B) C)
50 X座様 = 0 かつ Y座様 = 0 4.5 50 200/ローンを削除する ************************************	D) お化けのホ ーム部分

4-4 ドットとパワーフードの数をカウントしておきましょう

ゲームクリアの判定は、ドットをパワーフードをプレイヤーが全て食べたときで す。つまり、ドットの数=0かつパワーフードの数=0を判定する必要があります。 となると、存在するドットとパワーフードの数をつねに把握する必要があります。

考え方として、

- 1. 生成直前にカウントをOにする
- 2. クローンで生成するたびにカウントを1増やす
- 3. クローンが削除されるたびにカウントを1減らす

ということをしてあげれば良いです。

そこで、ドット、パワーフードそれぞれの数をカウントするための変数を作り、上記の仕組みを組み込んでください。

変数は全てのスプライト用でかまいません。



完成例は次のページです。



4-4 完成例

ドット

x座標を -180 、y座標を 120 にする 13 回繰り返す	ドット生成 ▼ を受け取ったとき	追加	
	x座標を -180 、y座標を 120 にする 13 回繰り返す		

クローンされたとき 表示する	追加	
ドットの数 ▼ を 1 ずつ変える		
もし パワーフード ▼ に触れた または プレイヤー ▼ に触れた または 色に触れた な6	追加	
ドットの数 ▼ を ・1 ずつ変える このクローンを削除する		
もし $x ص = 0$ かつ $y 座 = 0$ または $y 座 = 20$ なら	追加	
このクローンを削除する		
もし	追加	
ドットの数 ▼ を <mark>-1</mark> ずつ変える このクローンを削除する		

パワーフード

闷 が押されたとき	クローンされたとき	追加
	オハ9 つ パワーフードの数 ▼ を 1 ずつ変える	/
パワーフード生成 ● を受け取ったとき		追加
パワーフードの数 ▼ を 0 にする x座標を -180 、y座標を 120 にする		

4-5 スコア

- 変数「スコア」を作りましょう。
- 緑の旗が押されたときにスコアはOにしてください。どのスプライトに入れても 構いませんが、プロジェクト全体の中でひとつだけにしてください。
 (あちこちからスコアをOにする命令が動くと不具合発生の可能性があるため)
- ドットがプレイヤーに触れた時、ドットのクローンを削除してスコアを10増やしましょう。変数「ドットの数」も1減らしてください。
- パワーフードがプレイヤーに触れた時、パワーフードのクローンを削除してスコアを100加算するようにしましょう。変数「パワーフードの数」も1減らしてください。

プレイヤーがパワーフードの隣のドットに来たとき、パワーフードがプレイヤー に触れていないのに消えてしまう場合があります。そのときはパワーフードの大き さを少し小さめに調整してください。

動作確認では

設定した通りにスコアが増えることを確認してください。

変数「ドットの数」「パワーフードの数」についても、実際に見えている数と連動 していることを確認してください。

4-5 完成例

「スコアを0にする」をドットに入れました。他のスプライトでも構いませんが、 プロジェクトの中で1つだけにしましょう。



ドットがプレイヤーに触れた時にスコアを10増やす、ドットの数を1減らす、ク ローンを削除する。これは「クローンされた時」のコードに追加すると良いです。



パワーフード、プレイヤーに触れたらスコア100増やす、パワーフードの数を1減 らす、クローンを削除する。



4-6 クリア

①ステージの背景に下図のような「ステージ1クリア」を作ってください。



- ② ドットの数=0かつパワーフードの数=0の時、ステージを「ステージ1クリア」 にするコードを作ってください。
- ③ 今のままでは緑の旗がクリックされたときにも「ステージ1クリア」のままです。 緑の旗がクリックされた時に迷路の背景にしてください。

ここまでできたら、ドットとパワーフードがなくなった時に「ステージ1クリア」 になることを確認してください。

4-6 完成例

▶ が押されたとき	この例では「背景2」になって いますが、みなさんが迷路にし
背景を 背景2 ▼ にする パワーフード生成 ▼ を送る ずっと	ている背景のコスチューム名 にしてください。
もし ドットの数 = 0 かつ パワーフードの数 = 0 なら 背景を ステージ1クリア ▼ にする ペー・ペー・ペー・ペー・ペー・ペー・ペー・ペー・ペー・ペー・ペー・ペー・ペー・ペ	

【練習問題4-6】

- (1) クリアした途端に「ステージ1クリア」になってしまうので、1秒待ってから 「ステージ1クリア」にしてみましょう。
- (2) 「ステージ1クリア」になったら、プレイヤーを非表示にしましょう。 (答えに近いヒント) クリアになったことを知らせるメッセージを投げて、プレイヤーはそれを受け 取って消えるようにすると良いです。

ここまで問題なくできたら、変数「ドットの数」「パワーフードの数」は非表示に しましょう。

23

【練習問題4-6】完成例

(1) ステージ 🏴 が押されたとき 背景を 背景2 ▼ にする パワーフード生成 👻 を送る 追加 もし ドットの数 = 0 かつ パワーフードの数 = 0 なら 1 秒待つ 背景を ステージ1クリア マ にする • (2)ステージ 🍽 が押されたとき 背景を 背景2 ▼ にする パワーフード生成 👻 を送る もし ドットの数 = 0 かつ パワーフードの数 = 0 1 秒待つ 追加 背景を ステージ1クリア ▼ にする クリア 👻 を送る



5

【研究してみよう】

「クリアを送る」が下図の場所にあると、どのような動きになるか試してみましょう。

もし	ドットの豊	= 0	かつく	フーフ	数	= (なら
	クリア 🔹 २	を送る 👘					
	1 秒待つ						
背景を	ステージ1	クリア 👻	にする				
	ا و	· ·	r - 1				

4-7 微調整・不具合調整

4-7-1 ステージの表示が変

現段階で動作確認すると、クリアになった後にもう一度緑の旗を押すと最初は迷路が表示されていてドットが増えていくのですが、途中で「ステージ1クリア」の背景に変わってしまうことがあると思います。



原因は、ステージ。すこ~し、考えてみましょう。

▶ が押る	されたとき											
背景を(背景2 🔻 🕯	こする										
11		E成 🔹	を送る									
ずっと												
もし 🔇	ドットの	の数 =	0	〉かつ		パワー	フード	の数	= (0	》な	6
もし 🔇	ドット0 1 秒待・	D数 = っ ^い	0) かつ	• < (パワー	フード	の数) = (0	♪ な	6
もし 《 背景で	ドット0 1 秒待 ⁻ を ステージ	D数 = つ ・ ジ1クリ:	0 7 •	〉かつ ・ にする		パワー	フード 	の数 -) = (• •	》な	6
もし 《 背景で	ドットの 1 秒待 を ステージ クリア 、	D数 = つ ・ ジ1クリ: を送る		> かつ				の数 	-	0		6
もし 《 背景れ	ドット0 1 秒待 を ステージ クリア ・ チ	D数 = フ ジ1クリ: を送る		〉 かつ にする						0		6

これだと、いつでも「ドットの数=0かつパワーフードの数=0」の時にクリアに なるので、本来とは意図しないタイミングで「ステージ1クリア」になる危険性を含 んでいるようにみえます。

構造的なバグ(プログラムの欠陥)と言えるでしょう。

そこで、ドットの生成が終了したタイミングでドット(スプライト)が「ゲーム開始」のメッセージを送り、ゲーム中に起きる事象はそのメッセージを受け取って動くようにしましょう。

25

まずは、ドット生成終了後に「ゲーム開始を送る」を追加。

ドットのコード



ステージのコード



動作確認してください。

【ついでに】

現時点では、ドット生成中にもプレイヤーを動かすことができ、ドットを食べるこ とができてしまいます。

■チャレンジ

- (1) プレイヤーも、「ゲーム開始」を受け取ってから所定の位置に表示するように してください。
- (2) それだけだと、プレイヤーが出現する位置にもドットが生成され、ゲーム開始 時にプレイヤーが出現した途端その場所のドットが食われるため、得点が10 入ってしまいます。 ですので、そうならないようにゲーム開始時にも所定の位置にプレイヤーが表 示されるようにしてください。

🔋 📕 が押されたとき 🐨 🗤 🖓 🍋 A a a a a a a a a a a a a a a a a a a
x座標を 0 、y座標を -40 にする
-90 度に向ける
表示する
ゲーム開始 マ を受け取ったとき
x座標を 0 、y座標を -40 にする
-90 度に向ける
表示する。そこのため、ためのないないので、ないない
ತ್ರೆಕ್ರಿ
0.1 秒待つ
もし 左向き矢印 ▼ キーが押された なら
-90 度に向ける

完成例 プレイヤーのコード

4-7-2 壁をすり抜ける裏技ができてしまう

上矢印キーと左矢印キーを同時押しすると、プレイヤーは左上に動いて壁をすり 抜けてしまいます。

このように、斜めに動くようにキーを2つ同時押しすると意図していなかった不 具合が見られます。

これを防止するには、同時押しを働かせないプログラムに修正します。

「もし~なら」で四方向に動かす変わりに、「もし~なら・でなければ」を使います。 まず、「もし~なら・でなければ」を使い左矢印キーの部分を下図のようにしてく

ださい。



ドリームプラス https://drmpls.com

さらに、「でなければ」の中に「もし~なら・でなければ」を使って、上矢印キー と下矢印キーの時の動きを作ってください。



ここまでできたら、動作確認しましょう。 斜めに動かないこと、壁をすり抜けないことが確認できれば OK です。 パックマンもどき~迷路あり レベル1 https://scratch.mit.edu/projects/708192946

5 オバケを作る・プレイヤーとの連携

【オバケ概要】

迷路内を自由に自動で動く。

通常モードでは、プレイヤーと当たるとプレイヤーを殺す

プレイヤーがパワーフードを食べたら、一定時間青ざめモードになりプレイヤー と当たると食べられる。

プレイヤーに食べられると、点数の数字コスチュームになった後目玉になり、ホームへ戻る。

ホームへ戻った目玉は、一定時間経った後オバケに復活する。

5-1 スプライトを描く コスチュームを5つ用意





青ざめ

白

プレイヤーがパワーフードを食べたとき、一定時間こ のコスチュームになる。

「通常」を複製して作ると楽。

30

青ざめから通常に戻るとき、青ざめと白を点滅させる 「青ざめ」を複製して作ると楽。



いずれのコスチュームも例ですので、そのままじゃなくて構いませんが、場面に合わせて使えるように描いてください。

表示する大きさは、迷路の幅に収まる大きさにしてください。

5-2 動かすプログラミング

【チャレンジ】仕様

緑の旗が押された時、隠しましょう。

「ゲーム開始」を受け取ったら、表示しましょう。コスチュームは「通常」です。 表示する位置は x=0, y=0 です。

プレイヤーの時と同様に、迷路内を動くようにしましょう。 基本は 20 歩動かす。 壁に触れたら-20 歩動かすことにして動きを打ち消す。

向きを縦横に限定するため、90* [1~4の乱数] 度にしましょう。 (壁のないパックマンもどきで作ったことのある場合は振り返って参考にしてください。) オバケの見た目は回転しないようにしてください。

向きを決めた後、「20歩動かす」を?回繰り返しましょう。 ?の部分を乱数にすると、動く距離がランダムになります。 速く動きすぎる時は、これもプレイヤーの時と同様に待ち時間で調整しましょう。 プレイヤーは 0.1 秒の待ち時間を入れましたが、オバケも同じにするか増やすか 減らすかは実際に動かして調整してみましょう。

まずは自力で考えて作ってみましょう。

解答例は次のページに示しますが、コードの構成は解答例と違ってもかまいません。

仕様のとおりの動きが実現できれば OK です。

また、解答例が最善・最適のコードとは限りません。

もっと短いコード、効率の良いコード、わかりやすいコードでできればそれに越したことはありません。

そのように、誰かがもっと良い方法を発見・発明して、世の中は進んでいきます。

5-2【チャレンジ】解答例

· 🔁 が	押されたと	<u></u>							
隠す	- 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10								
ゲー.	ム開始 ▼	を受け取	ったとき	3					
x座標	を 0、	y座標を	0 6	こする					
コスヂ	ュームを	通常 🔻	にする						
表示す	-3								
ずっと									
	0.1 秒行	寺つ							
	0.1 秒f 90 *	寺つ 1 か	6 4) までの	の乱数	度	に向い	ける	
	0.1 秒f 90 * 1 から	寺つ 1 か ら 4	ら 4 までの話)まで(記数 [の乱数 回繰り) 度 返す	に向い	t3	
	0.1 秒1 90 · 1 か! 20 世	寺つ 1 かり ら 4 示動かす	ら 4 までのま)まで() 11数 [の乱数 回繰り) 度 返す	に向い	ける	
	0.1 秒1 90 * 1 か 20 歩	寺つ 1 か 6 4 示動かす) 色に触	ら 4 までのま nた)までの 乱数 [なら	の乱数) 度 返す	に 向 (- - - - - - - - - - - - -	ナる ・	
	0.1 秒1 90 * (1 から 20 歩	寺つ 1 か 5 4 手動かす)色に触れ 少動か	ら 4 までのき hた)までの 乱数 [なら	の乱数	と 変す い い		ナる ・ ・	
	0.1 秒1 90 * (1 か 20 歩	きつ 1 か う 4 示動かす) 色に触れ) 歩動か	ら (4) までのま れた す)までの 乱数 [なら	の乱数	と 度 返す ・	に向(・ ・	ナる ・ ・	
	0.1 秒1 90 * (1 か4 20 ± -20	きつ 1 か) 3 4 3 5 3 5 3 かす) 色に触れ) 歩動か	ら 4 までのま れた)までの 礼数 [なら	の乱数	と 度 返す ・ ・			
	0.1 秒1 90 * (1 から 20 歩	寺つ 1 か 1 か う 全に触れ う 少	6 4 までのま れた)までの 記数 [なら	の乱数	: 度 返す ・ ・ ・			

5-3 プレイヤーに触れた

【チャレンジ】仕様

(1) プレイヤーに触れたとき [通常] コスチュームなら、まずはシンプルに背景を ゲームオーバーにしてみましょう。

ステージにゲームオーバーの背景を作り、それを表示するようにしましょう。

(ヒント)メッセージを使います。

※注

後の段階で、プレイヤーに触れたとき [青ざめ] コスチュームならオバケがダメ ージを受けるコードを追加します。

ですので、この段階で [通常] コスチュームであることを判定するようにしてお きます。

5-3【チャレンジ】(1) 解答例

オバケ						
-20 歩動かす						
		٦				
もし プレイヤー ▼ に触れた	なら					
もし コスチュームの 名前 ▼		通常	なら			追加
ゲームオーバー 🔹 を送る	5			7		

ゲームオーバーになると、今はこんな感じです。



オバケもプレイヤーも動きます。

【チャレンジ】

(2) オバケが消えるようにしましょう。ドットもパワーフードも消えるようにしましょう。

34

(ヒント)ドットのパワーフードはクローンですので、クローンを削除します。

(3) クリアの時もオバケが消えるようにしましょう。

5-3 【チャレンジ】 (2) 解答例

オバケ





これで、ゲームオーバーの時は下図のようにシンプルになりました。



(3)



プログラム全体が動きつづけていますので、ステージのコードで「すべてを止める」 を追加しておきましょう。



5-4 食われた時のプレイヤー

【チャレンジ】仕様

(1)まずはシンプルにプレイヤーが消えるようにしましょう。

(2) プレイヤーの残り数を表す変数を作り、ゲーム開始時に2にして、プレイ ヤーがダメージを受けたら1ずつ減らし、0より小さくなったらゲームオーバ ーにしてみましょう。 プレイヤーはダメージを受けると一度消えて、2秒ほど待ってからゲーム再 開されるようにしてください。この時、オバケも消しましょう。

(ヒント)

プレイヤーダメージという変数を作りましょう。 「ゲームオーバー」を「プレイヤーダメージ」に差し替えるところもあります。

(3) プレイヤーの残り数を表すスプライトを作り、プレイヤーの残り数を表示し ましょう。



なお、この表示はゲームクリア時には非表示にしてください。

(4) 食われたら、徐々に薄くしていって消しましょう。

薄くなっている最中はキー操作で動いてはマズいので、他のスクリプトを止め ましょう。

(ヒント)

「幽霊の効果を20ずつ変える」を5回繰り返してから隠しましょう。

そのままだと「表示」にしても透明のままで見えませんので、ゲームスタート時には「幽霊の効果をOに」しましょう。

36

5-4 【チャレンジ】 解答例



(2)

プレイヤー命をゲーム開始時に2にする。プレイヤーに下記のコードを追加。 📔 が押されたとき x座標を 0 、y座標を -40 にする -90 度に向ける 追加 プレイヤー命 👻 を 🔰 にする 表示する

オバケのコードを変更



37

プレイヤーのコードに下記を追加



(3) プレイヤー残り数のコスチュームはプレイヤーを複製すると楽です。

プレイヤー残り数のコードの例。残り数分、クローンして表示します。





6 パワーフード関連のプログラム

【チャレンジ】仕様

- ●パワーフード
- (1)「オバケ青ざめ」メッセージを投げる

●オバケ

- (1)「オバケ青ざめ」メッセージを受け取ったらコスチュームを「青ざめ」へ変更。
- (2) 10秒間待つ。
- (3) インターバル 0.25 秒で「白」と「青ざめ」を入れ替えを10回繰り返す。
- (4) コスチュームを「通常」に変更。

(5) プレイヤーに当たった時にコスチュームが「青ざめ」か「白」だったら、スコ アを200増やす。その時、コスチュームを「200」にする。1 秒ほど待ってコス チュームを「目玉」にする。

(6) コスチュームが目玉になったら、オバケのホームエリアへ 0.5 秒で帰り、3 秒 待ってからコスチュームを通常へ戻す。

プログラムを見やすくするため、定義「帰る」を作ってそれを使ってみましょう。

6【チャレンジ】 解答例



●オバケ



0,		7			 		 	 		
もし プレイヤー 🗸	に触れたなら									
もし コスチュー	ムの 名前 ▼)=	通常	なら							
プレイヤーダ	メージ 👻 を送る									
隠す										
±1. 77	人の名前一	まざめ	**	+ / (-LO	之前	- 6	tre	5	
		, , ,					- (H			追加
	E 200 ずつ変える	5							-	5
1 秒待つ										
1	玉 🔹 にする									

この段階で動作確認すると、コスチュームが目玉になった後にも青ざめと白の点 滅になったり、通常コスチュームに戻ったりします。



その理由は、(1)~(4)のコレが無条件で動いているからです。

【チャレンジ】オバケ(7)

コスチュームの「青ざめ」「白」点滅~「通常」に戻る一連の流れを、コスチュー ムが「青ざめ」の時だけ動くように修正してください。

6【チャレンジ】解答例 オバケ(6)

1 秒待つ コスチュームを 目玉 • にする	
は し コスチュームの 名前 ▼ = 目玉 なら 帰る し 、 、 、 、 、 、 、 、 、 、 、 、 、	追加
定義 帰る	定義「帰る」作成
0.5 秒でx座標を 0 に、y座標を 0 に変える 3 秒待つ コスチュームを 通常 → にする	

オバケ(7) オパケ青ざめ ▼ を受け取ったとき コスチュームを 青ざめ 🔻 にする 10 秒待つ この条件を追加 = 青ざめ なら もし 🧹 コスチュームの 名前 🔻 10 回繰り返す コスチュームを 白・ にする 0.25 秒待つ コスチュームを 青ざめ 🔹 にする 0.25 秒待つ コスチュームを 通常 🗸 にする

レベル2で行う予定のこと

オバケを4匹に増やす ステージを増やす ボーナスフルーツを出す 音も入れる?