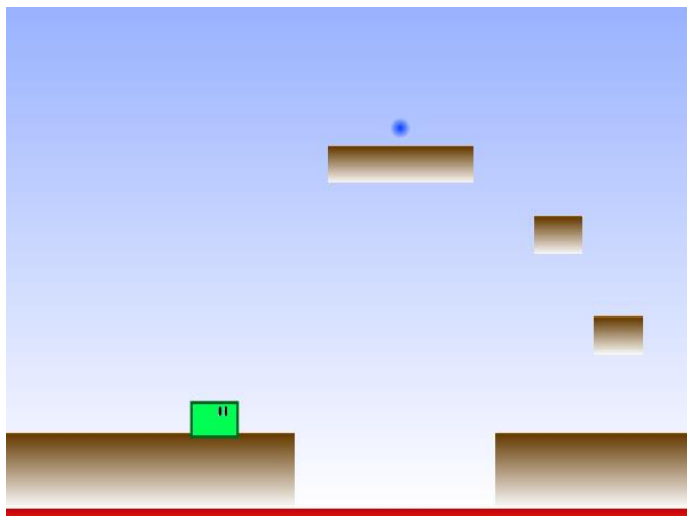


スライムの冒険



○こんなゲームです

- ▶ スライムをコントロールしながらお宝を取ってステージを進めてください
- ▶ どんな背景（プラットフォームと呼びます）を作るかが、このゲームの面白みです
- ▶ 難易度を上げながら楽しいゲームにしてください

スプライトを用意しよう

自分で描いてもライブラリから持ってきててもOKです



【スライム】
左右の矢印キーとスペースキーで操作



【お宝】
4つゲットするとクリア

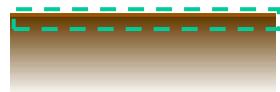
豆テスト

○ジャンプのコードを書いてください

- ▶スペースキーを押したら
- ▶40飛び上がって
- ▶40落下する

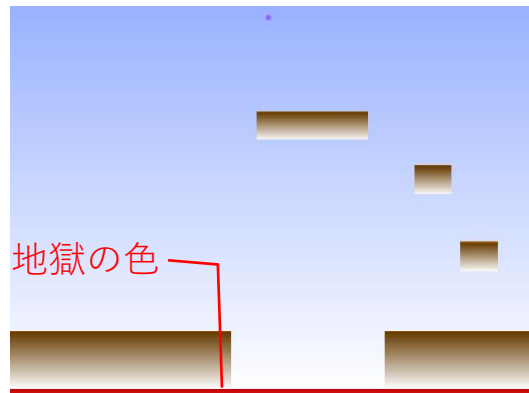
ステージを作る

上の部分は全画面で
同じ色に統一

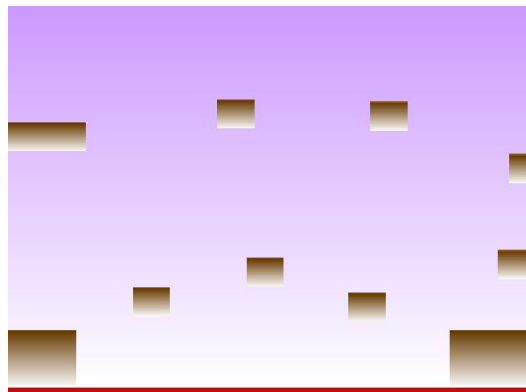


プラットフォームの
上の部分は色判定で
よくわかるはっきり
した色を使ってください

○背景とゲームオーバーを描く



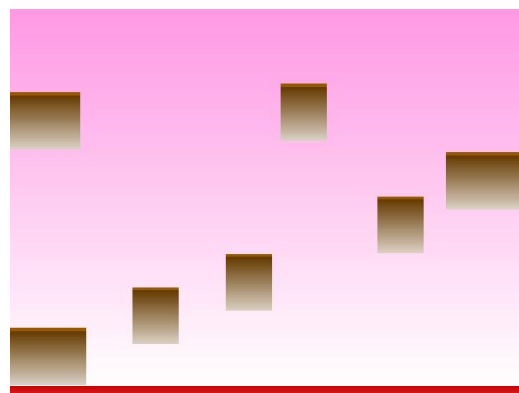
背景1



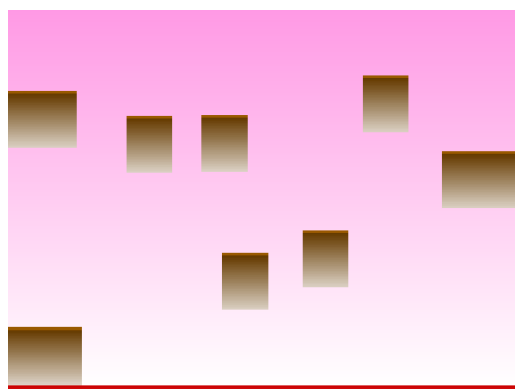
背景2



クリア



背景3

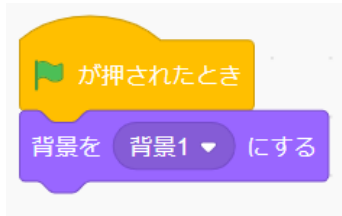


背景4

背景1と背景2をクリア
した後、背景3と背景4が
交互に出現する「地獄の谷」
に行きます

ステージのプログラム

○初期画面



○お宝から画面切替のメッセージが送られてくると



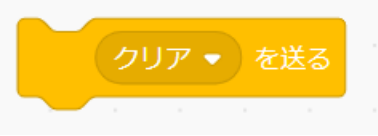
メッセージ発信
(お宝)

メッセージ受信
(ステージのプログラム)

背景1をクリアすると



背景2をクリアすると



○「地獄の谷」のプログラム

背景3をクリアすると



スライムを作る



スライムのプログラム ①



○コスチュームを描きます
(右向きと左向き)



○左右に動くとき
スライムの向きを変えたい

最初のスポーン位置

The script starts with a 'when green flag clicked' event. It sets the x-coordinate to -200 and the y-coordinate to -60, then rotates 90 degrees. A 'forever' loop follows, containing two 'if' conditions. The first 'if' condition checks for the right arrow key; if pressed, it sets the costume to 'right-facing' and moves the x-coordinate 8 units right. The second 'if' condition checks for the left arrow key; if pressed, it sets the costume to 'left-facing' and moves the x-coordinate 8 units left.

=

The final script is a single 'when green flag clicked' event block. It sets the x-coordinate to -200 and the y-coordinate to -60, then rotates 90 degrees. It then enters a 'forever' loop. Inside the loop, there are two 'if' conditions. The first 'if' condition checks for the right arrow key; if pressed, it sets the costume to 'right-facing' and moves the x-coordinate 8 units right. The second 'if' condition checks for the left arrow key; if pressed, it sets the costume to 'left-facing' and moves the x-coordinate 8 units left.

スライムのプログラム ②



○プラットフォームの上でジャンプ
(そうでないときは落下)

```
が押されたとき
ずっと
  ↓
```

```
もし 色に触れた ではない なら
  y座標を -10 ずつ変える
でなければ
```

プラットフォームの色

```
もし スペース キーが押された なら
  10 回繰り返す
  y座標を 10 ずつ変える
  ↓
```



=

```
が押されたとき
ずっと
  もし 色に触れた ではない なら
    y座標を -10 ずつ変える
  でなければ
    もし スペース キーが押された なら
      10 回繰り返す
      y座標を 10 ずつ変える
    ↓
  ↓
```

落下する

プラットフォームの上でジャンプする



○画面の下（地獄の色）にふれると…

が押されたとき

地獄の色

ずっと

もし 色に触れた なら

x座標を -200 、y座標を -60 にする

90 度に向ける

最初のスポーン位置

が押されたとき

ずっと

もし 色に触れた なら

x座標を -200 、y座標を -60 にする

90 度に向ける

ステージ2 を受け取ったとき

x座標を -200 、y座標を -60 にする

90 度に向ける

ステージ3 を受け取ったとき

x座標を -200 、y座標を -60 にする

90 度に向ける

クリア を受け取ったとき

x座標を 0 、y座標を -60 にする

90 度に向ける

背景2、背景3、クリア画面のスポーン位置は自由に変えてください

お宝のプログラム



お宝のスポーン位置は自由に変えてください

○スライムがお宝をゲットすると次のステージ（背景）に進みます

お宝のスポーン位置

```
が押されたとき
x座標を 36 、y座標を 95 にする
表示する
```

背景1

```
スライム に触れた まで待つ
ステージ3 を送る
x座標を -220 、y座標を 113 にする
```

背景2クリアで背景3に進む

```
スライム に触れた まで待つ
ステージ2 を送る
x座標を -207 、y座標を 80 にする
```

背景1クリアで背景2に進む

```
スライム に触れた まで待つ
隠す
クリア を送る
```

背景3&4をクリアするとゲームクリア

```
が押されたとき
x座標を 36 、y座標を 95 にする
表示する
スライム に触れた まで待つ
ステージ2 を送る
x座標を -207 、y座標を 80 にする
スライム に触れた まで待つ
ステージ3 を送る
x座標を -220 、y座標を 113 にする
スライム に触れた まで待つ
隠す
クリア を送る
```

○レベルチェック

- 必要なブロックを見つけられる
- 見本を見ながらブロックを並べられる
- 間違ったところを自分で見つけられる、直せる
- ブロックの意味（コマンド）がなんとなく分かる
- ペイントエディタが使える（初歩レベル）
- 制作途中で作品を上書き保存できる

○関連する学習

x座標、y座標

正の数、負の数

順次処理

衝突判定

条件分岐

メッセージ

コスチュームの切り替え

ステージの切り替え

ライブラリの活用

BGM・音の入れ方

デバッグ

キーボード操作