

情報科学演習3

(6) 状態遷移図

脱出ゲーム

- まずはサンプルを実行してみよう
 - Game1, Game2 をウェブからダウンロード

複雑なスクリプトを作る

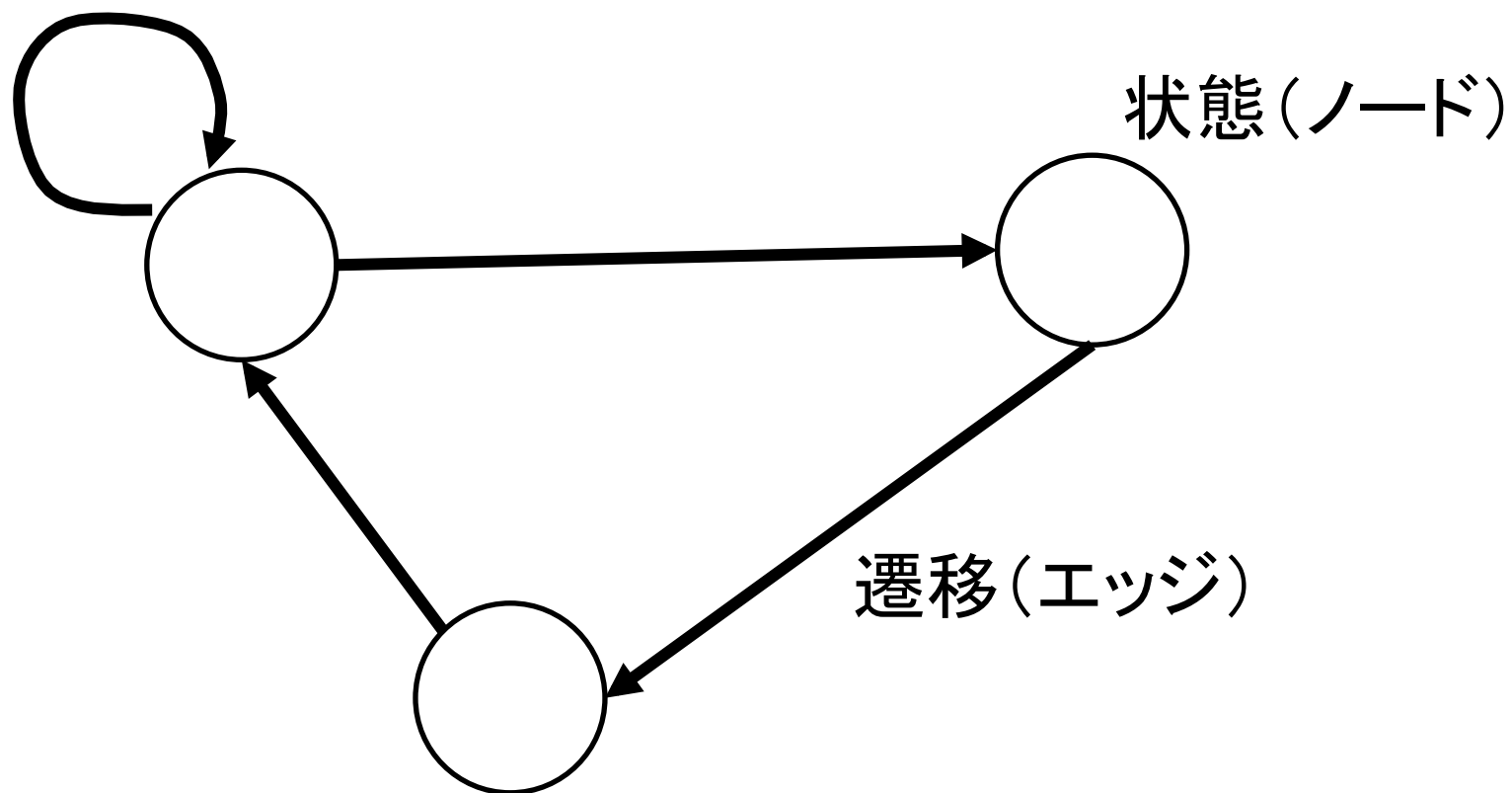
1. 設計

- 構想を練る
- スクリプトに近い設計図を作る
(状態遷移図・フラグ管理)

2. 実装

- 絵(スプライト)を作成
- スクリプトを作成

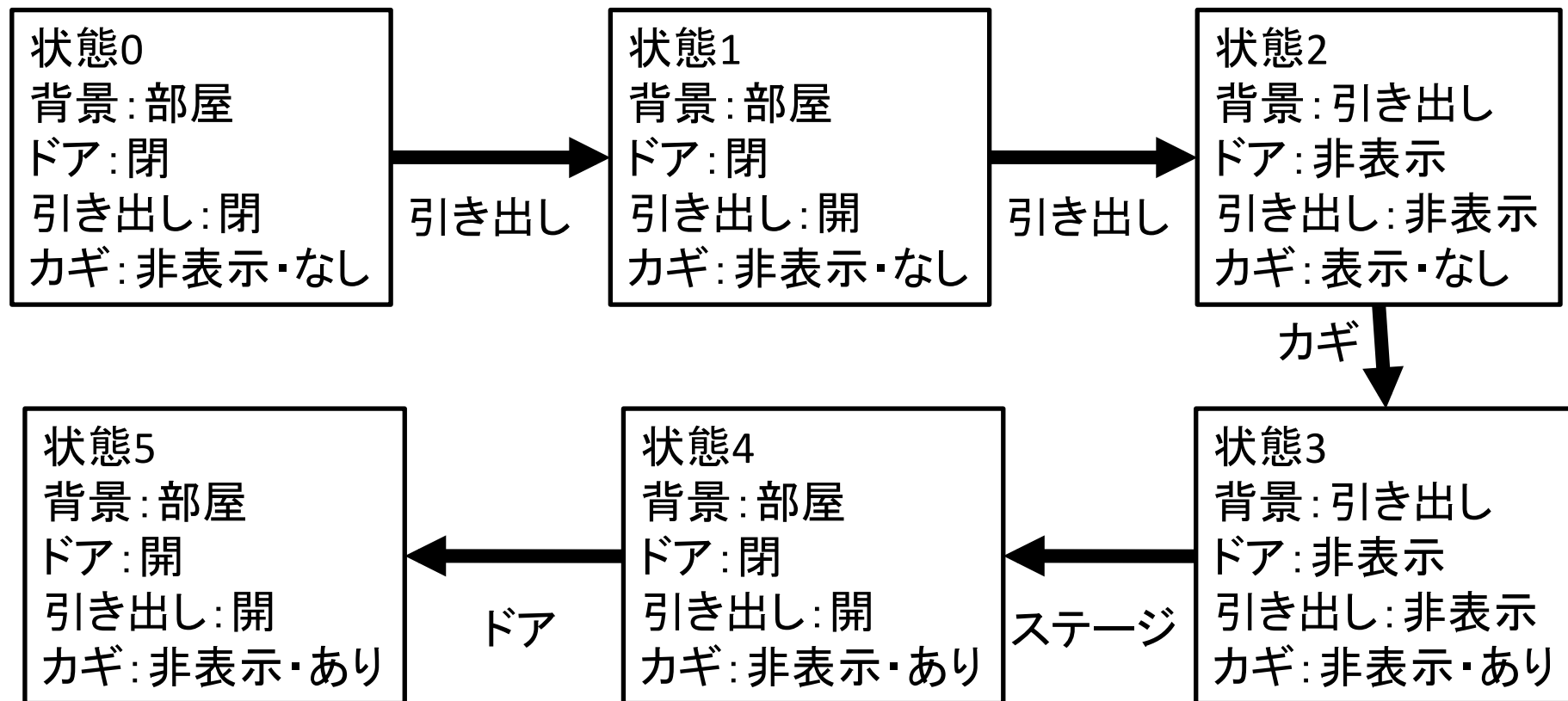
状態遷移図



状態遷移図の作り方

- 可能な状態をすべて書きだす
- 各状態で可能な行動をすべて書きだす
(矢印の元)
- 各行動の結果の遷移先をつなぐ
(矢印の先)

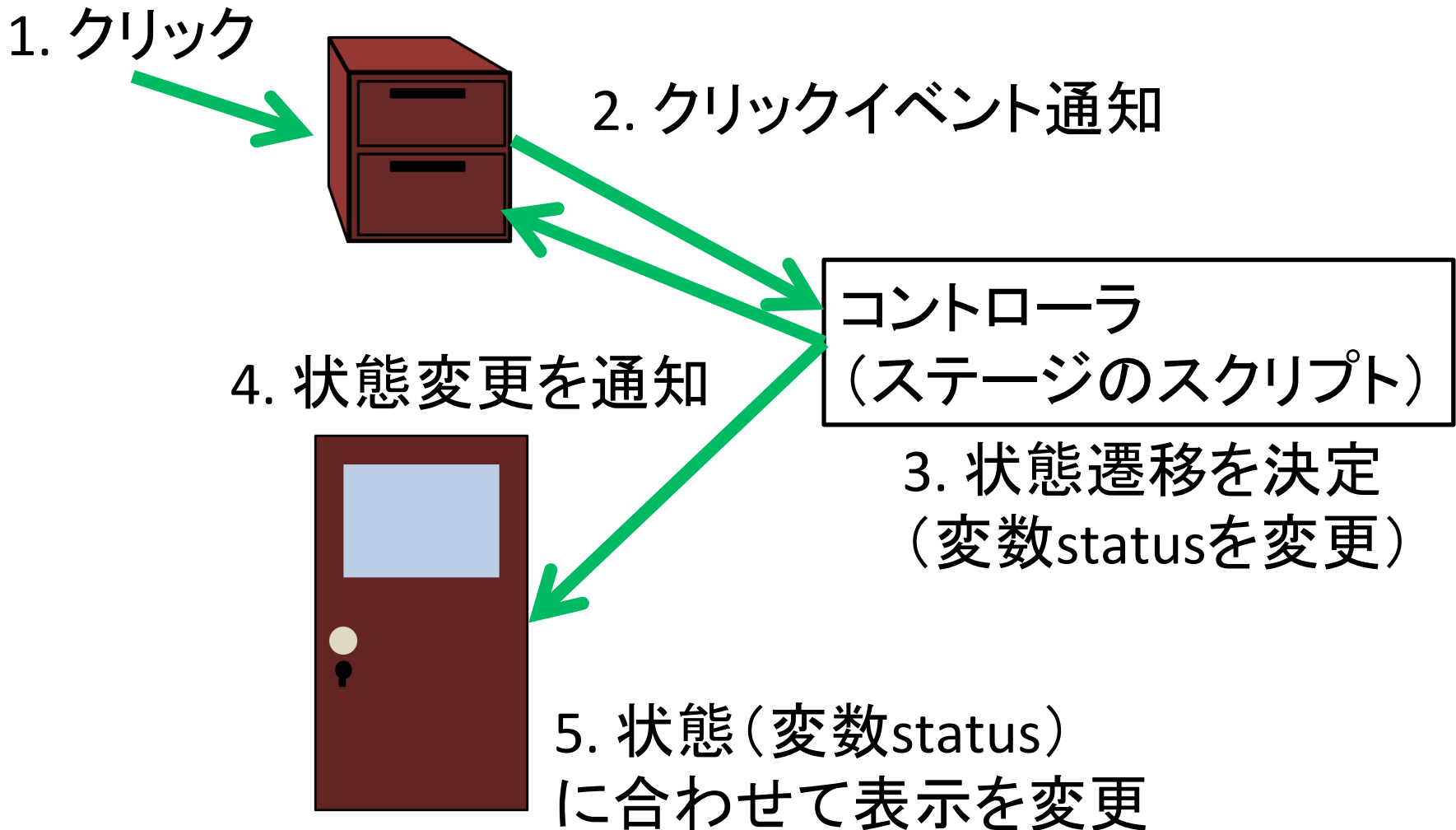
Game1の状態遷移図



状態遷移図からの実装

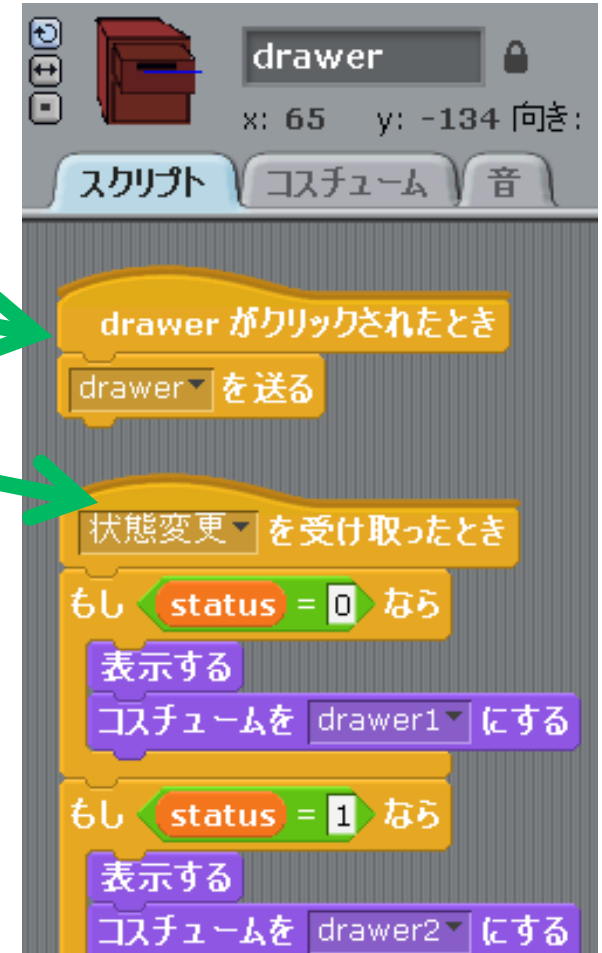
- グローバル変数「status」を作る
(現在の状態を表す)
- ステージに背景を作成
- スプライトで部品を作成
 - クリックできる部分になる(ドアノブ・スイッチ等)
- スクリプトを作成
 - 部品のスクリプト
 - ステージのスクリプト

全体の動作



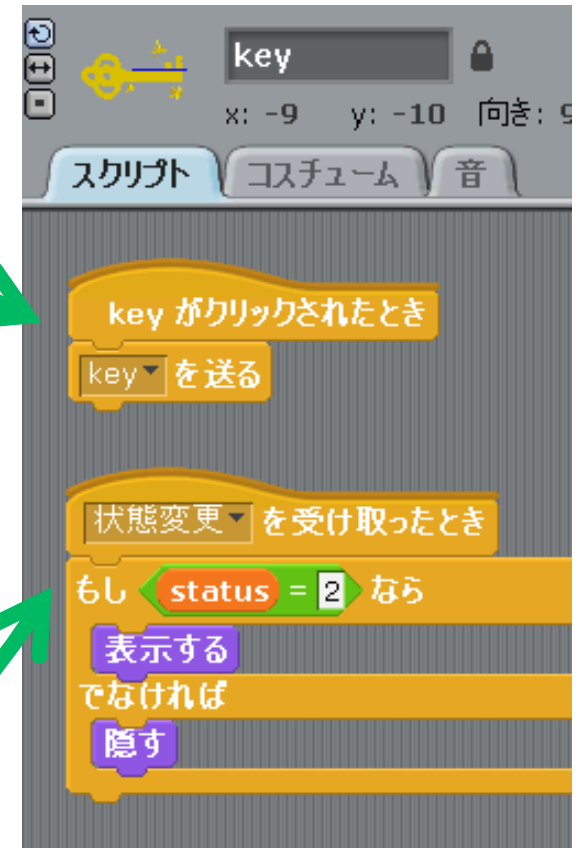
各スプライトのスク립ト

- 2つのイベントに対応
 1. スプライトがクリックされたとき
 - メッセージを送る
(コントローラが処理する)
 2. 「状態変更」を受け取ったとき
 - 変数 status に応じて表示を変更
 - status = 0 のときは初期化



各スプライトのスク립ト

- 2つのイベントに対応
 1. スプライトがクリックされたとき
 - メッセージを送る
(コントローラが処理する)
 2. 「状態変更」を受け取ったとき
 - 変数 status に応じて表示を変更
 - 条件はまとめて判定してもよい



コントローラのスクリプト(ステージ)

- 基本的な動作
 1. 変数statusを更新
 2. 「状態変更」を通知
- 3種類のイベントに対応
 - 旗がクリックされたとき
 - 変数statusを初期化して通知
 - 状態変更を受け取ったとき
 - 状態に合わせて背景を変更
 - 部品クリックが通知されたとき
 - 基本的な動作(サンプルの通り)

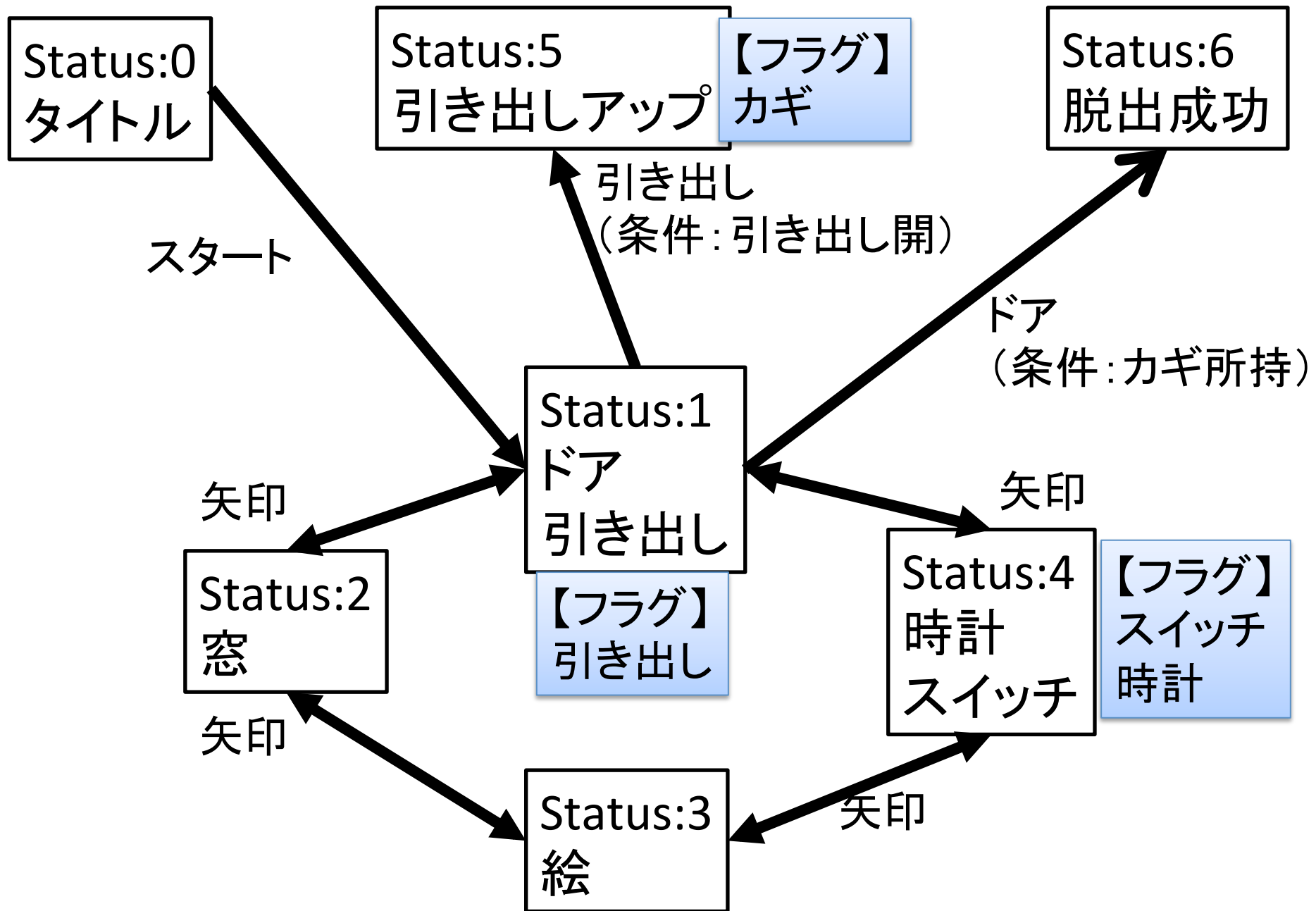


フラグ管理

- さらに複雑なスクリプトを作るとき、状態遷移だけで設計すると状態数が多くなりすぎる
→ 多くの状態に関する要素は変数で管理
- フラグの例
 - カギについてのフラグ
持っていない(変数が0)、持っている(変数が1)

Game2の設計

- フラグ
 - 引き出し = 0 or 1 (閉 or 開)
 - カギ = 0 or 1 (未所持 or 所持)
 - スイッチ = 0 or 1 (OFF or ON)
 - 時計 = 0 or 1 (左 or 右)
- 状態遷移図は次のスライド



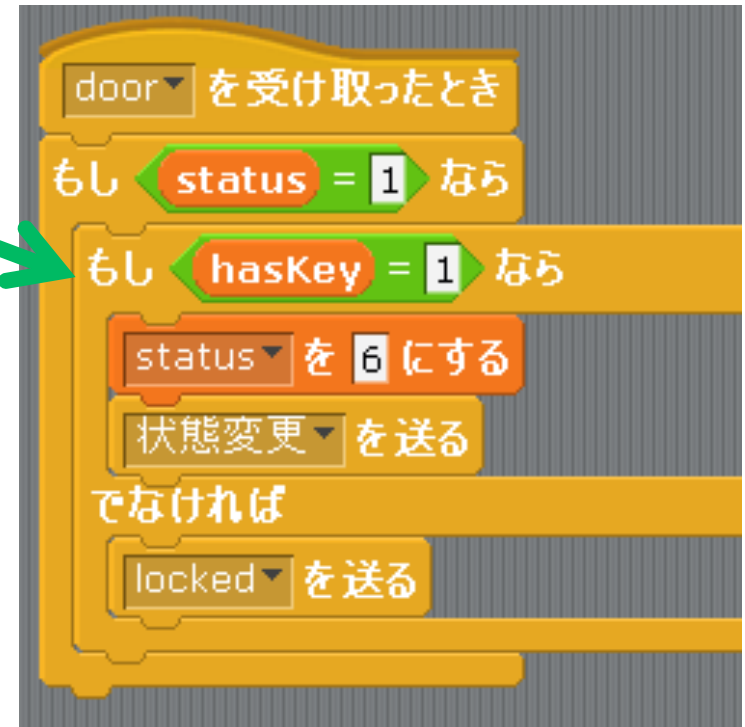
コントローラでのフラグ管理

- 旗がクリックされたとき
 - 各フラグを初期化する
- 部品クリックが通知された時
 - フラグ変数の値を変更する



コントローラでのフラグ管理(つづき)

- 状態とフラグに基づいて遷移を決定する



各スプライトのスク립ト

- 状態とフラグに合わせて表示を更新



課題

- 脱出ゲームのための状態遷移図を作成する
 - 紙で提出(手書き or PCで作って印刷)
 - 来週の授業で回収するので、提出用コピーを用意しておくこと
- 要件
 - 状態数: 6以上
 - 部品数: 2以上
 - フラグは使わなくてもよい