

# デジタルコンテンツ:最終レポート

(株)C.I.

5418007 内村方哉 5418034 古田瑛啓 5418035 前田真志

2020 年 8 月 25 日

## 1 作品アイデア・狙い

このゲームの対象ユーザは学生で、学生というのは良くも悪くも退屈な人間が多い。朝は早く起き、学校に登校し、授業を受けてまた家に帰る。そのようなルーティンを繰り返している学生にとって日常に組み込めるようなゲームを考案する。退屈な授業、つぶしようなない休み時間などにできるようなものを提案する。また、難易度を選択、ゲームを自身でできるようにした。

## 2 プログラムの仕様

### 2.1 donkama クラス

- init メソッド ゲームシステムで利用する変数を初期化するメソッド
- now メソッド ノーツを表示するメソッド
- tap メソッド キーボードをたたく処理をするメソッド
- miss メソッド ミスしたときのカウントメソッド
- gameOver メソッド ゲーム終了時の処理をするメソッド

### 2.2 Tekuteku クラス

tekuteku クラスでは、主にゲームの1つである tekuteku のゲーム制御をするクラスである。

- init メソッド ゲームシステムで利用する変数を初期化するメソッド
- loadStage メソッド ゲームで利用するマップをテキストファイルを読み込んでステージを生成するメソッド
- turn メソッド 移動する自機を上下左右に制御するメソッド
- showStage メソッド loadStage メソッドで読み込んだステージを表示するメソッド
- showBlock メソッド 各マスごとのマップを表示するメソッド
- move メソッド 自機が移動している様子を表示するメソッド
- movedProcessing メソッド 自機が移動後におけるマスごとの処理をするメソッド
- miss メソッド 自機がマップ外に入った際に残機を計算するメソッド

- gameOver メソッド 残機が0になったときにゲームを終了するのを制御するメソッド

## 2.3 palot について

- timeKeeper メソッド 制限時間をカウントする
- init メソッド 変数, 画像初期化宣言する
- drawRect メソッド 画像を表示する
- drawHouse メソッド 家を表示する
- move メソッド 画像を動かす
- judgeRect メソッド 画像が家に入ったか判定する
- liveonPalotCount メソッド 家に入っている鳥の数を数える
- palotBack メソッド 家から鳥を開放する
- keyPressed メソッド キーボードから選択肢を選択するプログラム
- answerScreen メソッド 答えを画面に表示する
- answerJudge メソッド 答えがあっているか判定する
- resultScreen メソッド 結果を画面に表示
- mouseReleased メソッド 選択肢をクリックして離れたときを判定する
- setup メソッド 初期化する
- draw メソッド 描画する

## 2.4 その他

- loadButton メソッド ボタンで利用する画像を読み込むメソッド
- setup メソッド 初期化などの初期設定に必要なメソッドを呼び出すメソッド
- draw メソッド ゲームシステムの画面遷移をおこなうメソッド
- viewScore メソッド スコアを表示するメソッド
- donkamaSet メソッド donkama がゲームプレイで選択されたときに利用するメソッド
- printScene メソッド デバック用
- showDiffHover メソッド ボタンを押したときの画像を表示するメソッド
- showEndHover メソッド ボタンを押したときの画像を表示するメソッド
- startScreen メソッド 最初の画面を表示するメソッド
- diffScreen メソッド 難易度画面を表示するメソッド
- endScreen メソッド ゲーム終了画面を表示するメソッド
- mouseReleased メソッド マウスの制御に関するメソッド
- keyPressed メソッド キーボードの入力に関するメソッド

### 3 プログラム実行の様子

#### 3.1 メイン画面



図 1 メイン画面

プログラムをまず実行すると、このような画面が現れる。三つのゲームから一つを選択する。

#### 3.2 難易度を選択画面

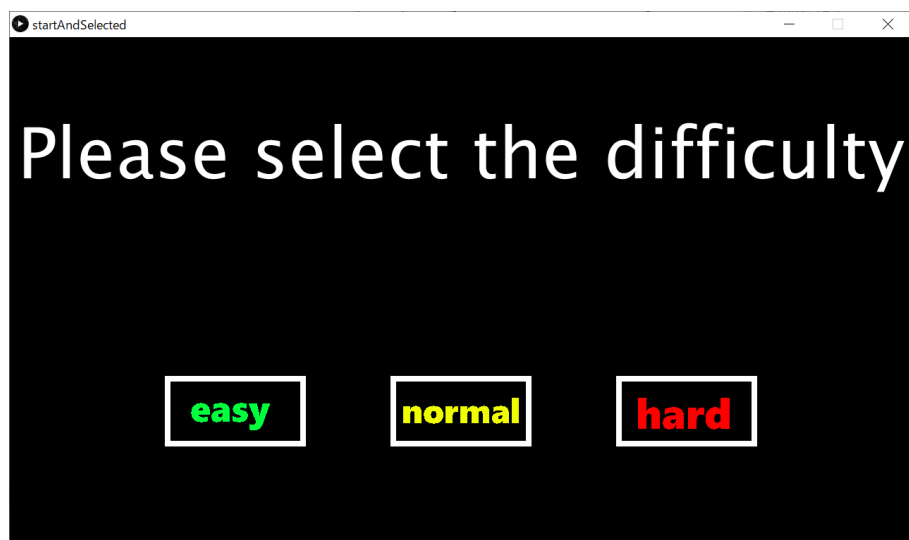


図 2 難易度選択画面

ゲームを選択すると、難易度選択画面が現れる。easy, normal, hard の三つの難易度から選択する。

### 3.3 てくてく

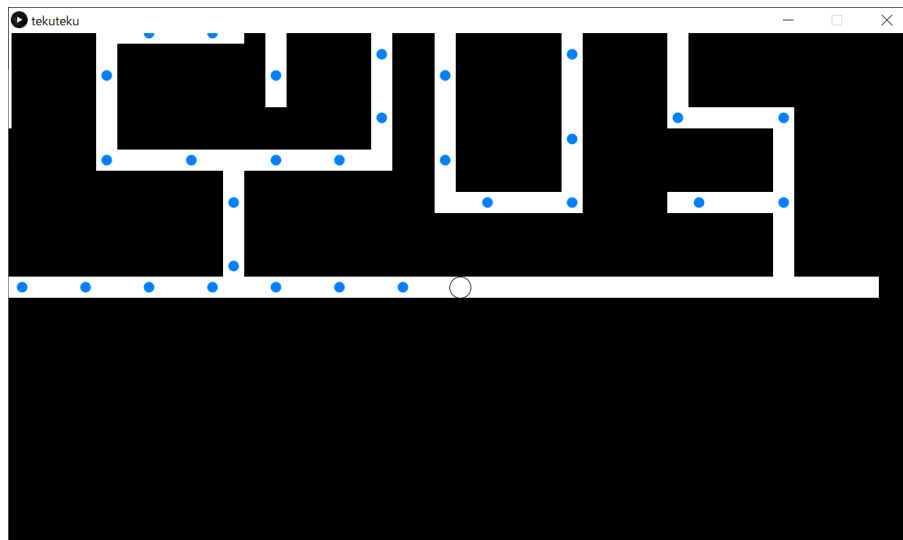


図 3 てくてくの実行画面

てくてくを選択すると、このような画面が現れる．自動で移動する自分のキャラクターを方向キーで操作して、マップからコースアウトしないようにしながらアイテムを集めてゴールする．正確な操作が求められる．

### 3.4 ドンカマチック

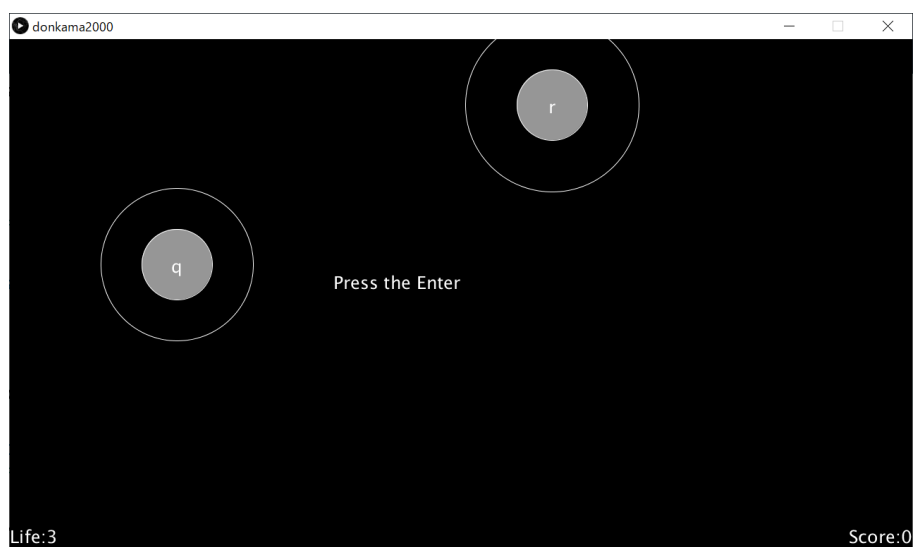


図 4 ドンカマの実行画面

ドンカマチックを選択すると、このような画面が現れる．円の外側から縮まってくる円と円がちょうど重なるように指定されたキーボードをたたいて得点を稼ぐ．ライフポイントが 0 になったらゲームオーバーとなる．

る．リズム感が鍛えられる．

### 3.5 おうちに何羽いるでしょうゲーム

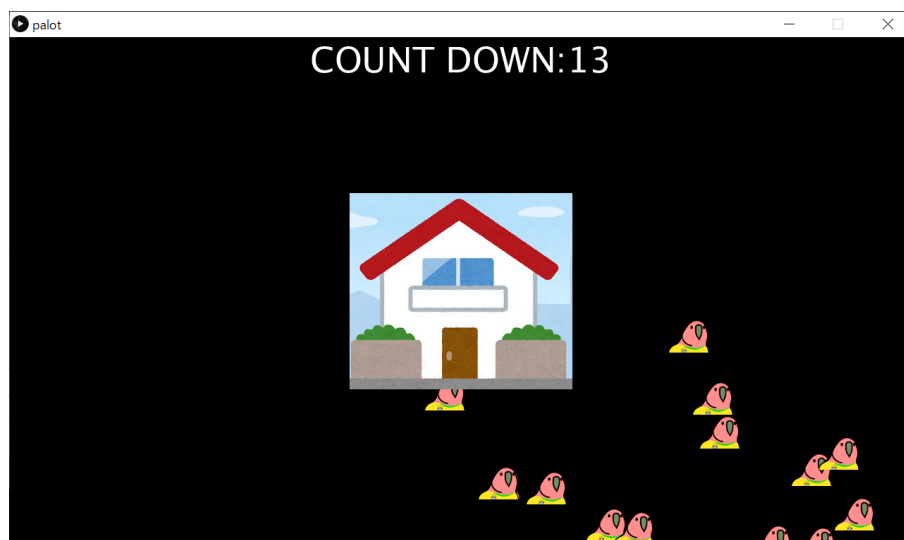


図 5 ドンカマの実行画面

おうちに何羽いるでしょうゲームを選択するとこのような画面が現れる．随時移動している鳥が家の中に入る．家の中に何羽いるのかを答えよう．回答方式は三択．記憶力が鍛えられます．

## 4 仕様運用

当初の想定ของเกมの動きはできた．ただ，ミニゲームが3つあるうちの一つだけがゲームシステムに組み込めなかったのは残念だった．ゲームの選択ボタンは色が違う画像を生成して正しい処理を行うことによって，あたかもボタンを推したようなエフェクトを実現できた．

## 5 発表時のコメント

- ゲーム開始直前に間がなく，すぐに始まってしまうため準備時間もしくは合図が欲しい
- てくてくゲームのマップ画面の生成の仕方がうまい．
- ドンカマチックでどのボタンを推せばいいかもっと詳しく明示してほしい
- 難易度 easy が easy になっていない

## 6 今後の課題

まず，ゲームシステムを完成まで行くことができなかった．原因として当初予定としていた計画に対してメンバーのタスクをこなすスキルが一致していなく，メンバー自身も個人的な課題やタスクが多く重なっていたためゲーム制作難易度とメンバーのスキルが一致していなかった．対策として，ゲームを制作する際にか

かる時間コストを一度見直す必要があった。またオブジェクト指向プログラミングの授業でコードをクラス化することを習っているはずだが、クラスごとに pde ファイルに分けていなくコードの保守性やメンテナンス性に欠けていた部分があったので、実際の作成でもこの考え方を利用すべきだった。さらに、発表時のコメントでも言われたように全体としての完成度が低かったため、もっと時間をかけるべき作成物であった。ゲーム自体の作成はもともと構成段階で考えていた部分をコードで実行することは難しかったが、実際にコードを作成している部分は考える要素を多かったので楽しい部分もすこしあった。