グループ課題

2020/12/04 遠山

# 概要

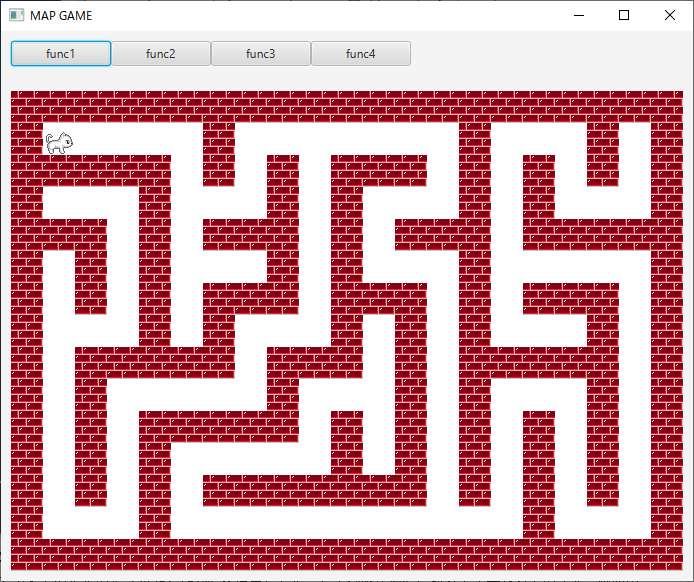
この資料ではグループ課題について説明します．みなさんには5名程度のグループで課題に取り組んでもらい，その成果を中間・最終発表会で披露してもらいます．ベースとなるプログラムは提供しますが，みなさんの力でそれを発展させ，オリジナルなゲームへと昇華させてください．グループメンバーは教員が適宜決めました．メンバー表は授業資料の中に入っていますので確認してください．

グループ課題ですから，グループ内で意思疎通をすることが重要になります．特に，話し合いながら少しずつプログラムを読んだり，わからないところを確認したりするプロセスを経験していただくことが重要です．作品の完成だけを目指す進め方は評価されません．例えば，得意な人にすべてやってもらうのは論外です．もちろん，分業することや，わからないところを教え合うことはかまいませんが，どんな独自機能を作るか議論したり，それぞれの作業でできたものを組み合わせて動かしたり，組み合わせの際のバグを取り除いたりといった結合の部分が最も重要です．このため，課題自体も複数名で取り組んだ方がうまくいくように作成してあります．

また，グループの独自性を出すために，他のグループがどのような作品を仕上げようとしているのかについてサーベイすることも有効です．公式なサーベイチャンスとしては，中間発表会が挙げられます．

グループ課題の評価は，発表会での教員・TA・聴講者全員の投票結果が反映されます．プロダクトの完成度のみでなく，発表の質も求められます．発表は5分程度ですので，短時間でアピールできるように，「デバッグモード」や「デモモード」(予め設定した初期値で動作する)などを実装することも有効です．どのように発表でアピールするかを検討することも，グループ課題に含みます．

課題では，ゴールに向けて白猫を動かすゲームを作ってもらいます．課題で用いているのは前回の課題で学んでいただいたJavaFX（FXMLを使用）です．できる限りMVCに則って実装されています．画面のイメージは以下です．白猫がいる部分の正方形を，以下では便宜的に「1マス」と呼びます．



　下記にこのゲームでできることを示しました．ただし，あえて未実装としてある部分もあります．グループで下記を実装しつつ，上記には設定されていない特別な機能も実装して，他のグループとは異なるできるだけ面白いゲームを作ってください．

* キーボードのhjkl（←↓↑→）で猫が移動する
* 猫は左上からスタートする
* 猫が特定のアイテムを取得していればゴールできる
  + 現状ではアイテムが何もない，ゴールもない
* 猫がゴールすると新しい地図が生成される
  + 現状では新しい地図が生成されない
* ユーザがfunc1～func4のキーを押すとゲーム上で何らかのスペシャルイベントが発生する
  + 現状ではスペシャルイベントは発生しない

# 実行方法

　MS Teamsから以下のファイルをダウンロードしてください．すべて同じディレクトリ内に置いてください．

* MapGame.java
* MapData.java
* MoveChara.java
* MapGameController.java
* MapGame.fxml

　以下のようにコンパイル・実行して動作を確認してください（javafxのパスは，以前の授業資料を参考にして確認するなどしてください）．

$ javac –-module-path (javafxのパス) –-add-modules javafx.controls,javafx.fxml \*.java

$ java –-module-path (javafxのパス) –-add-modules javafx.controls,javafx.fxml MapGame

課題には「必須機能を手順に従って行う」部分と「独自機能を作り込む」部分とがあります． 必須機能は5つ，独自機能は2つ以上実現してください．必須機能が適切に実現されていること，興味深い独自機能が多く実現されていること，発表会でのトーク・デモ・質疑応答がうまくできていることが，それぞれ高評価となります．必要に応じて，JDB（Javaのデバッガ），Scene Builder（JavaFXのGUIを構築するためのGUI環境）[[1]](#footnote-1)などを使用してもかまいません．

# 必須課題：5つの機能の実現

## 設計・実装・テスト

システム開発世界では，仕様書の作成・設計書の作成・実装（コーディング）・テストの実施，という作業を通じて，システムを完成させていきます．本課題でも，実際のシステム開発の世界で行われている仕様書の作成・設計書の作成・実装・テストの実施という仕事を体験していただきます．

* **仕様**：ユーザが「こういう機能があったらいいな」という目線で語ったことを文章や図で表したもの
* **設計**：どのようにして仕様で求められている機能を作るのかを決めること
* **実装**：設計に沿ってプログラムを作ること
* **テスト**：プログラムが期待された通りに動くこと，期待された通りには動かないことを確認すること

仕様書の

作成

設計書の

作成

実装

テストの

実施

5つの機能はグループ内の別の人が**実装**を担当してください（5名以上のグループにしてありますので，1人1機能分担できるはずです．もし不測の事態で分担が難しくなった場合は教員へ連絡してください）．実装以外にも，**設計**と**テスト**という仕事があります．設計・実装・テストはそれぞれ別の人が担当しなければなりません．これは，1つの機能を完成させる過程に複数の人を関与させることで，1人では気付かなかった勘違いを取り除いたり，1人では気付かなかったより良い実現の方法に気付いたりするという，協調の利点を生かすためです．システム開発現場でもこのように，あえて異なる人が同じ機能を確認することは多々あります．例えば以下のように分担する方法が考えられます．

|  |  |  |  |
| --- | --- | --- | --- |
| 機能名 | 設計 | 実装 | テスト |
| 必須機能1 | Aさん | Eさん | Dさん |
| 必須機能2 | Bさん | Aさん | Eさん |
| 必須機能3 | Cさん | Bさん | Aさん |
| 必須機能4 | Dさん | Cさん | Bさん |
| 必須機能5 | Eさん | Dさん | Cさん |
| 独自機能1 | Aさん | Eさん | Dさん |
| 独自機能2 | Bさん | Aさん | Eさん |

以下に，設計・実装・テストの作業内容を示します．

* 設計
  + どのような機能を作成するのかについて文章で表現する．いつ，どこへ，何を，どのようにするかが明らかになるように，数字を用いて記述するとよい．主語と述語の対応が取れているか確認すること．主語は，ユーザの操作なのか，キャラクターの動きなのか，プログラム内でのイベント発生なのかなど，多様になる可能性がある．1文を短くするほうが，主述の対応がとりやすくなる．文で表現することが難しい場合は図表を用いてもよい．
* 実装
  + 作成したクラスやメソッドや機能はどのような機能を持っているのか，それらのクラスやメソッドや機能は他のどのクラスやメソッドとどのように関係しているのかをコメントに書く（コメントはjavadoc形式で作成する）．
* テスト
  + 設計で定められた入力とは異なるものが入力されても問題ないか，キャラクターに普通ではない動きをさせてもバグが生じないか等，イレギュラーなケースをできる限り多く想定し，それらを全て試す．テスト項目を考える際には，設計を読むことが有益に働く場合がある．

## 設計・実装・テストの例

以下に，本課題の白猫が移動する機能の**設計**と**テスト**の例を示しますので参考にしてください．実装についてはMapGameController.java（52～96行目付近）, MoveChara.javaなどのソースコードを確認してください．

* **設計**
  + 左へ進む機能
    - ユーザがhキーを押したとき，白猫が現在いるマスよりも1マス左へ移動する．ただし，左隣のマスが壁の場合は，白猫は移動しない．
    - 左へ移動しようとする際には，白猫が左を向いた画像に切り替える．また，pngディレクトリにある左向きの画像3枚を0.5秒おきに切り替えて，白猫が足踏みをしているように見せる．
  + 右へ進む機能
    - ユーザがlキーを押したとき，白猫が現在いるマスよりも1マス右へ移動する．ただし，右隣のマスが壁の場合は，白猫は移動しない．
    - 右へ移動しようとする際には，白猫が右を向いた画像に切り替える．また，pngディレクトリにある右向きの画像3枚を0.5秒おきに切り替えて，白猫が足踏みをしているように見せる．
  + 上へ進む機能
    - ユーザがkキーを押したとき，白猫が現在いるマスよりも1マス上へ移動する．ただし，上隣のマスが壁の場合は，白猫は移動しない．
    - 上へ移動しようとする際には，白猫が上を向いた画像に切り替える．また，pngディレクトリにある上向きの画像3枚を0.5秒おきに切り替えて，白猫が足踏みをしているように見せる．
  + 下へ進む機能
    - ユーザがjキーを押したとき，白猫が現在いるマスよりも1マス下へ移動する．ただし，下隣のマスが壁の場合は，白猫は移動しない．
    - 下へ移動しようとする際には，白猫が下を向いた画像に切り替える．また，pngディレクトリにある下向きの画像3枚を0.5秒おきに切り替えて，白猫が足踏みをしているように見せる．
* **テスト**
  + 左へ進む機能
    - 白猫が現在いるマスの左隣のマスが空白のとき，ユーザがhキーを押したら，白猫が現在いるマスよりも1マス左へ移動することを確認する．
    - 白猫が現在いるマスの左隣が壁のとき，ユーザがhキーを押しても白猫がどこへも移動しないことを確認する．
    - 左へ移動しようとする際に，白猫が左を向いた画像が表示されていることを確認する．
    - 左へ移動しようとする際に，白猫が左を向いた画像が0.5秒間隔で3枚を切り替えて表示されていることを確認する．
  + 右へ進む機能
    - 白猫が現在いるマスの右隣のマスが空白のとき，ユーザがlキーを押したら，白猫が現在いるマスよりも1マス右へ移動することを確認する．
    - 白猫が現在いるマスの右隣が壁のとき，ユーザがlキーを押しても白猫がどこへも移動しないことを確認する．
    - 右へ移動しようとする際に，白猫が右を向いた画像が表示されていることを確認する．
    - 右へ移動しようとする際に，白猫が右を向いた画像が0.5秒間隔で3枚を切り替えて表示されていることを確認する．
  + 上へ進む機能
    - 白猫が現在いるマスの上隣のマスが空白のとき，ユーザがkキーを押したら，白猫が現在いるマスよりも1マス上へ移動することを確認する．
    - 白猫が現在いるマスの上隣が壁のとき，ユーザがkキーを押しても白猫がどこへも移動しないことを確認する．
    - 上へ移動しようとする際に，白猫が上を向いた画像が表示されていることを確認する．
    - 上へ移動しようとする際に，白猫が上を向いた画像が0.5秒間隔で3枚を切り替えて表示されていることを確認する．
  + 下へ進む機能
    - 白猫が現在いるマスの下隣のマスが空白のとき，ユーザがjキーを押したら，白猫が現在いるマスよりも1マス下へ移動することを確認する．
    - 白猫が現在いるマスの下隣が壁のとき，ユーザがjキーを押しても白猫がどこへも移動しないことを確認する．
    - 下へ移動しようとする際に，白猫が下を向いた画像が表示されていることを確認する．
    - 下へ移動しようとする際に，白猫が下を向いた画像が0.5秒間隔で3枚を切り替えて表示されていることを確認する．

　以下では，グループで実現してもらいたい5つの必須機能について，**仕様**を示します．仕様に示されたユーザの願いをくみ取って，設計を行い，設計を参考にして実装し，実装し終えたものをテストしてください．なお，必須機能間で情報交換をしなければうまく進められないところが数多くあります．「誰がどこまでの仕事をする」といった役割分担にこだわらないで積極的に情報交換を行うことが重要です．自分の担当機能だけを作ることではなく，機能を結合したときにきちんと動くことが最優先です．

## 必須機能1：アイテムの配置

仕様：白猫がゲットできるようなアイテムをいくつか迷路に置いてほしい．アイテムは何でも良いが，アイテムをゲットするとゲーム上で何かが起こってほしい．取ったアイテムは，画面上に表示されるようにしてほしい．アイテムが1つではつまらないので，ひとつのマップに3つくらい置いてほしい．

ヒント：MapData.java にプログラムを記述する．アイテム画像は32x32ピクセルで用意し，pngディレクトリに入れる．アイテムはLabelを使って配置する．アイテム取得後はsetText等で画面に表示する．

## 必須機能2：キャラクターのゴール

仕様：白猫がゴールの場所に到着しただけでは，ゲームクリアにならないようにしてほしい．白猫が指定されたアイテムをゲットし終えているときに，ゴールの場所へ行くと，ゲームクリアになるようにしてほしい．

## 必須機能3：クリア後新マップの生成

仕様：ゲームクリアしたら，前回とはまったく異なる新しいマップが自動的に作られて，次のゲームができるようにしてほしい．

## 必須機能4：スコアのカウント

仕様：白猫が特定の動きをしたらスコアを加点したり，減点したりする機能を付けて欲しい．

## 必須機能5：BGMや効果音の設定

　ゲーム中はBGMを流してほしい．白猫がアイテムを取った時や，ゲームをクリアした時などに効果音を流してほしい．効果音はゲームを楽しめるものであればどんなものでもよいが，著作権などの問題にひっかからない，フリーで使用できるものにしてほしい．

※MS Teamsの「サンプルプログラム」フォルダに，BGMを流すサンプルプログラムがあります．

# 必須課題：独自機能の実現

　他のグループにない機能を最低2つ考えて実現してください．必須機能と同じように，設計・実装・テストは別々の人が行ってください．また，それぞれの独自機能の実装は異なる人が担当してください．

# 今後のスケジュール

　以下に今後のスケジュールを示します．ここに示したスケジュールは，最低限のスケジュールです．ここに書かれているよりも早く進めることをお勧めします．特に，個別のプログラムはうまく動いても，結合するとうまく動かなくなる，ということはシステム開発の世界で頻繁に起こることです．1月中旬までに全体を作り終えるような気持ちで進めてください．なお，下記とは異なる順序や方法で進めたい場合もあると思いますので，その際には教員まで相談してください．

## 1.準備

12/4は各グループで，互いの得意なこと，好きなこと，経験してきたことなどについて自己紹介を行いながら，以下の1～6（ただし5の最終発表会は当日に誰が当たるかわからないので除外）で課題提出を行う人の分担表を作成し，分担表を代表者1名が提出してください．代表者とは「提出」を行う代表者なので必要に応じて催促などしてください．課題自体は全員で分担して責任をもって進めていただきます．以下の分担表例のように，本日も含めると今後計6回，グループで課題提出を行う機会があります．1人につき1回程度，グループ課題を提出していただく計算になります．5名グループの場合は，すみませんが誰かが2回担当してください．

＜分担表＞（コピーして使用してください）

|  |  |  |
| --- | --- | --- |
| 通し番号 | スケジュール | グループ課題提出者（例） |
| 1 | 1.準備 | Aさん |
| 2 | 2.中間発表会 | Bさん |
| 3 | 3.個別の必須機能のプログラム提出 | Cさん |
| 4 | 4. 必須機能のみのゲームの提出 | Dさん |
| 5 | 5. 独自機能の設計・テストの提出 | Eさん |
| 6 | 7. 発表会後 | Fさん |

## 2.中間発表会

12/18に予定している中間発表会では，グループ内の役割分担と，各必須機能の設計とテストについて日本語で記述したもの（実装つまりプログラムはまだなくてよい），今後製作予定の独自機能の仕様を紹介してもらいます．5分程度の動画（この授業の動画資料のように，スライドを提示する方法でかまいません）にまとめてください．詳しくは追って指示します．

## 3.個別の必須機能のプログラム提出

　1/8までに，必須機能5つそれぞれについて，テストまで完了した状態のプログラムを提出してください．この時点では，ベースとなるプログラムに対して開発した必須機能1つだけを加えると，その1つだけがうまく動くことが確認できる状態でかまいません（他の人が開発した必須機能を組み合わせるとうまく動かなくなるという状態でもOKとします）．

## 4.必須機能のみのゲームの提出

　1/15までに，必須機能5つを結合してひとつのゲームとして動くようになったもの（プログラムとfxml）をグループで1セット提出してください．

## 5.独自機能の設計・テストの提出

　1/22までに，2つ以上の独自機能の設計とテストを日本語で記述したものを，グループで1セット提出してください．なお，中間発表会を踏まえて方針を変更することも可能とします．

## 6.最終発表会

1/28の最終発表会は遠隔リアルタイム（8:40～11:50）で行います．作成したゲームの特徴を紹介し，特徴が際立つようなデモを行ってください．また，工夫した点などを説明してください．発表者や発表順は当日発表しますので，授業開始時点から全員が参加していること，全員が発表できるように準備しておくことが重要です．また，発表の際に，画面共有の操作を手伝うといったことは，発表者以外の人が行ってかまいません．

## 7.発表会後

発表会終了後は，発表会当日に得られた情報も踏まえて，個人およびグループでレポートを作成し，グループでとりまとめて提出してください．内容は本資料末尾を参考にしてください．

# 注意事項

　画像や音楽などの素材はどのようなものを使用してもかまいません．ただし，素材を提供している人が示している利用規約を遵守してください．たとえば「いらすとや」では，規約の範囲内であれば個人・法人，商用・非商用問わず無料で素材を使用することができますが，規約の範囲外であれば著作者に問い合わせをする必要があります．参考（いらすとや利用規約）　<https://www.irasutoya.com/p/terms.html>

注意しなければならないのは，同じように無料で使用することができる素材であっても，利用範囲や利用目的は作者の意図によって異なる点です．場合によっては，著作物の出処情報を併記する必要があったり，特定の目的に限って使用しなければならなかったりします．無料で使用できる場合でも，使い方によっては使用者側の責任が問われる場合がありますので，必ず利用規約を確認してください．

本グループ課題で，誰かが作成してくれた何らかの著作物を使用する場合は，学校での教育目的で使用すること，特定された相手のみに閉じた状態で公開されること，私的・非商用利用であることがポイントになります．もし著作物の使用に関して，利用規約を読んでも迷うことがあれば，教員やTAへ連絡してください．

なお，無料で提供されているアプリなどのプログラムについても，ライセンス事項が定められています．UbuntuやJDKにも定められています．このあたりの総論は授業動画にも示していますので確認してください．

# 今後の提出物

以下のレポートは期末に提出してもらいますが，どの機能は誰が実装担当なのかなどの役割分担や，プログラムの設計などについては，プログラムを作り始める前に決めることです．これらは12/18中間発表で示していただきます．当然ですが，虚偽申告があった場合には相応の処分となります．

## 個人レポート

* グループの自分以外の各メンバーがグループ課題の完成に向けてどのようなことをしたのか，1人ひとりについて記しましょう．代表して先生に質問した，グループメンバー間のケンカを仲裁した，オンラインで話し合いをすることを提案した，などの調整に関わる部分でもかまいません．
* 他のグループの発表会を聴いて疑問に思ったこと，素晴らしいと思ったことについて，それぞれ最低１点ずつ記しましょう
* 自分達のグループの進め方を振り返り，どのような点を改善すると今後もっとよい進め方ができると考えられるかを記しましょう
* 以上について書いたものをグループの代表者へ提出してください．提出はグループの代表者1名に行っていただきます．

## グループレポート（代表者1名が提出してください）

* 作品のプログラム（fxmlも含む）
* どのような独自機能を作成したかの説明
* 5つの必須機能と2つ以上の独自機能についての役割分担表（設計/実装/テスト）※以下は例

|  |  |  |  |
| --- | --- | --- | --- |
| 機能名 | 設計 | 実装 | テスト |
| 必須機能1 | Aさん | Eさん | Dさん |
| 必須機能2 | Bさん | Aさん | Eさん |
| 必須機能3 | Cさん | Bさん | Aさん |
| 必須機能4 | Dさん | Cさん | Bさん |
| 必須機能5 | Eさん | Dさん | Cさん |
| 独自機能1 | Aさん | Eさん | Dさん |
| 独自機能2 | Bさん | Aさん | Eさん |

* 5つの必須機能と2つ以上の独自機能について作成した仕様の文章，プログラム処理の内容を説明したもの（クラスやメソッドを作った場合はその名前も明記する），テスト項目の文章
* 実行結果
  + PNG形式の画像と，このゲームで遊ぼうとする人（大学生程度を想定）へのごく簡単な事前説明
* グループメンバーの個人レポート
  + メンバー全員から集めた個人レポートをそのまま貼り付ける
* 参考文献
  + 今回の課題を進めるに当たって参考にした書籍，webサイトのURL，人の発言などを列挙
* 謝辞
  + グループ以外の人で，課題達成までに特に協力してくれた人がいれば明記

1. Gluon Scene Builderなどがあります．https://gluonhq.com/products/scene-builder/ [↑](#footnote-ref-1)