

目次

1. 要求仕様書	... P2
2. スパイラル開発記録	... P4
2.1. SP1	... P4
2.2. SP2	... P8
2.3. SP3	... P15
2.4. 最終 (SP4)	... P21

要求仕様書

1. 全体

1-1. 課題プログラム作成の背景

2 年後期に開講される「ソフトウェア開発論」の課題として作成する Java プログラムである。

1-2. 課題の目的

自由課題を作成することでプログラミングの理解を深める。

2. 課題作成の前提条件

2-1. 課題作成における制約条件

Java で開発したプログラムであること。

クラス数 3, 行数 200 行以上のプログラムであること。

2-2. 学則上の制約条件

(1) 必ず本人が作成したプログラムであること

(2) 独自性は必ずしも必要ではないが、ソースコードは作成者本人のオリジナルであること。

(3) 参考にしたサイト, 書籍, プログラム等あれば, 必ず参考資料として明記すること。
ない場合は 不正行為とみなされる場合がある。

3. 課題の要件 (説明書)

3-1. 概要

本ゲームは、迷路状の街を効率よく巡ってクーポン券を使用し、いくら得をできるかを競うゲームである。ゲームは 1 日目の 6:00 からスタートし、4 日目の 6:00 になると終了する。また、ゲーム開始時の所持金は 20000 円であり、この所持金がなくなると制限時間を待たずしてゲームが終了する。

3-2. 詳細

3-2-1. ゲーム開始

S キーを押すとゲームが開始する。

3-2-2. ゲーム中

(1) 画面について

左側の画面には町の様子が表示され、(以下ゲーム画面と呼称) 右側の画面にはクーポン券や現在時刻などの情報が表示される (以下クーポン画面と呼称)。

ゲーム画面上の黄色の三角形はプレイヤーを表しており、その最も鋭利な角がプレイヤーの向いている方向を示してる。また、ゲーム画面上の黒色

の太線は壁を表しており、プレイヤーはこれを超えることはできない。ゲーム画面上の家マークは店を表しており、マーク上には店名が記されている。また、マークの色はその店のイメージカラーを示している。

クーポン画面上部の水色のエリアには、現在時刻、残金、得した金額が表示される。また、画面下部には最大5つのクーポン券が表示される。クーポン券には店名、対象商品名、通常価格、割引価格、有効期限が記載されている。また、クーポン券の色はその店のイメージカラーによって決まっている。

(2) 操作方法について

矢印キーを押すことにより、該当する方向へプレイヤーを動かすことができる。また、町は計9枚の迷路によって構成されており、表示されている迷路外へ出ると隣の迷路へ移ることができる。さらに、店とプレイヤーが接触した状態で、クーポン画面上の該当するクーポン券をクリックするとクーポン券を使用することができる。

(3) クーポン券について

クーポン券は一定間隔でランダムに生成され、使用した、もしくは有効期限が切れた場合は自動的に画面から消える。また、クーポン券はその店の営業時間内でのみ使用可能であり、営業時間外では店とプレイヤーが接触した状態でクーポン画面上の該当するクーポン券をクリックしても使用することはできない。各店の営業時間は以下のようである。

RAWSON	24 時間営業
モクドナルド	24 時間営業
ガスト	24 時間営業
吉田家	24 時間営業
ミニマムバリュー	24 時間営業
TOHU シネマズ	8 時～22 時
ムーンボックス	8 時～23 時
マツモトキヨコ	9 時～23 時
INUQLO	10 時～20 時
YAMATA 電機	10 時～20 時
ミトリ	11 時～20 時

また、クーポン券を使用した場合はその内容に応じて購入や飲食にかかった時間が経過する。

3-2-3. ゲーム終了

得した金額がスコアとして表示される。

S キーを押すと再プレイできる。

ゲームを終える場合はウィンドウの×をクリックする。

スパイラル開発記録

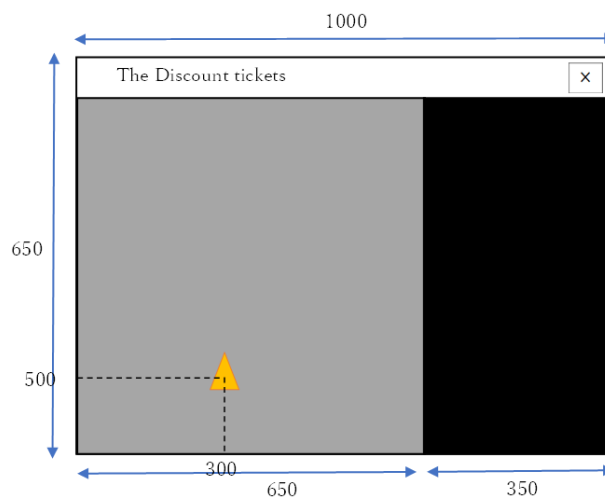
—SP1—

<SP1 要件定義>

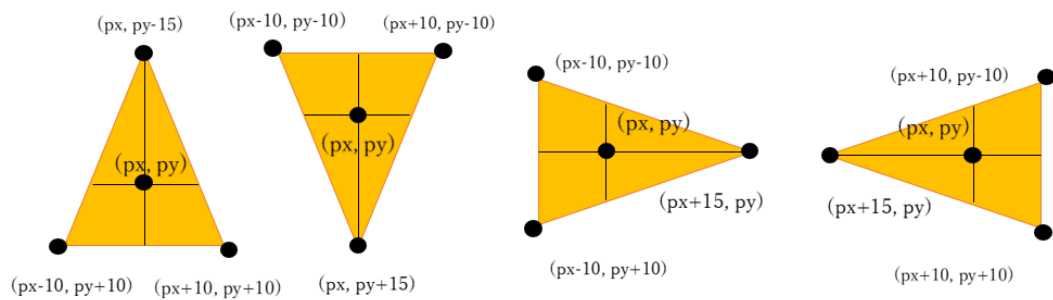
- ウィンドウを表示し，迷路画面とクーポン画面を配置する
 - ・ 起動するとウィンドウが表示される
 - ・ 大きさは 1000×650
 - ・ 位置は (0, 0)
 - ・ 閉じるボタンを押すとプログラムが終了する
 - ・ 左側に迷路画面を表示，大きさは 650×650，背景色はグレー
 - ・ 右側にクーポン画面を表示，大きさは 350×650，背景色は黒
- プレーヤーを矢印キーで操作できる
 - ・ 最初は迷路画面の下側に上を向いて表示される.
 - ・ クーポン画面には表示されない
 - ・ プレーヤーは黄色い三角形
 - ・ 1度矢印キーを押すとその向きを向く
 - ・ 続けて同じ方向の矢印キーを押すとその方向に進む
 - ・ 違う方向の矢印キーを押すとその方向を向く

<SP1 外部設計>

- ウィンドウ



●プレイヤー



<SP1 内部設計>

●5つのクラス

●TheDiscountTickets

- ・JFrame, メインクラス
- ・MazePanel を乗せる

●GameManager

- ・Thread
- ・Player を持つ
- ・一定時間ごとに Player を移動する
- ・MazePanel を再描画する

●Player

- ・ゲーム管理者に移動される
- ・画面の再描画時に描画される
- ・mazePanel のキーイベントを受け取るキーリスナー

●MazePanal

- ・JPanel
- ・迷路画面
- ・再描画時に, GameManager を描画する

●TicketaPanel

- ・Jpanel
- ・クーポン画面

●スレッド構造

●メインスレッド

- ・ イベント処理のループ
- ・ 画面の描画
- ・ 閉じるボタンで終了

●GameManager スレッド

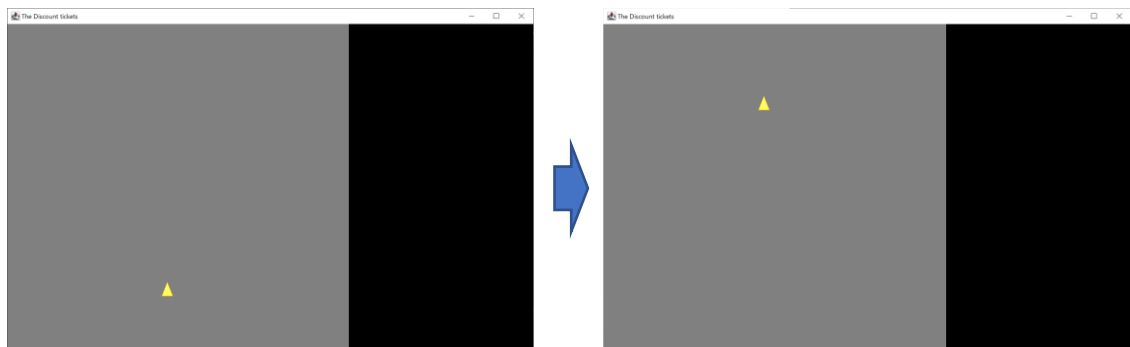
- ・ ゲーム自体のループ
- ・ 時間の経過とともにプレイヤーを移動
- ・ 終了しない（無限ループ）

●キーイベントによるプレイヤーの移動

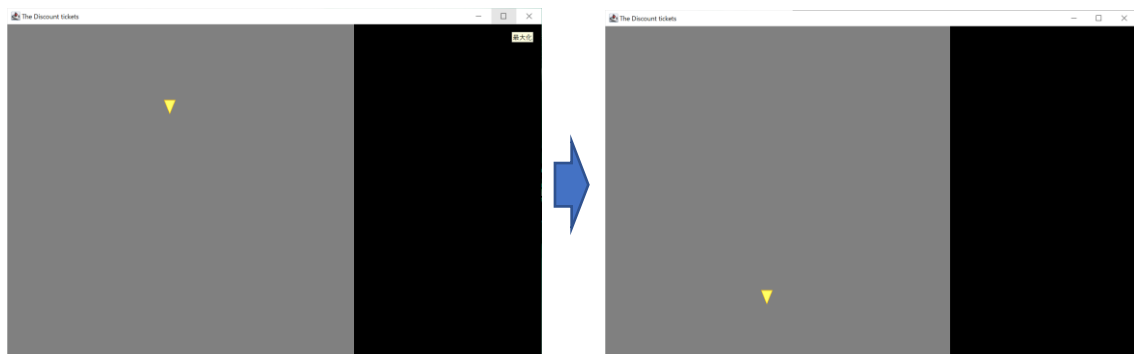
- ・ 上矢印キーに 0，下矢印キーに 1，右矢印キーに 2，左矢印キーに 3 の数字を割り当てる
- ・ 現在の方向を記憶する int 型変数 dir に格納されている値と押された矢印キーの値を比較する．同じであればプレイヤーの移動量を表す double 型変数 dx, dy を更新し，異なれば dir を更新する．

<SP1 の進捗>

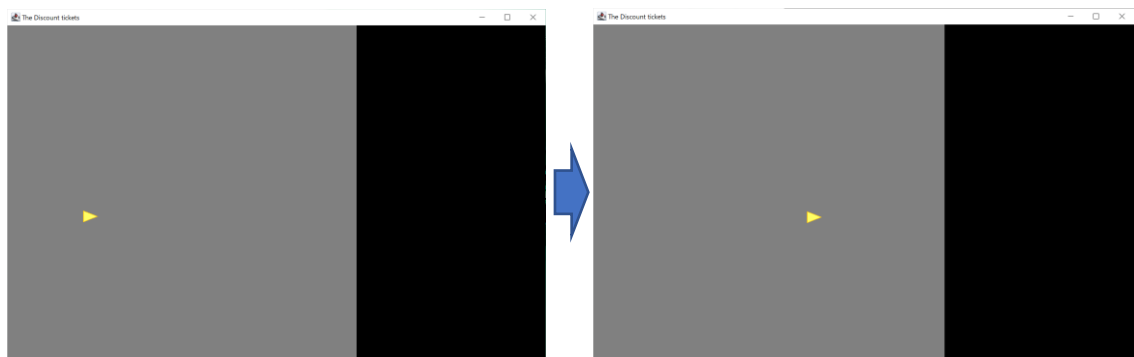
- ・ 上向き移動



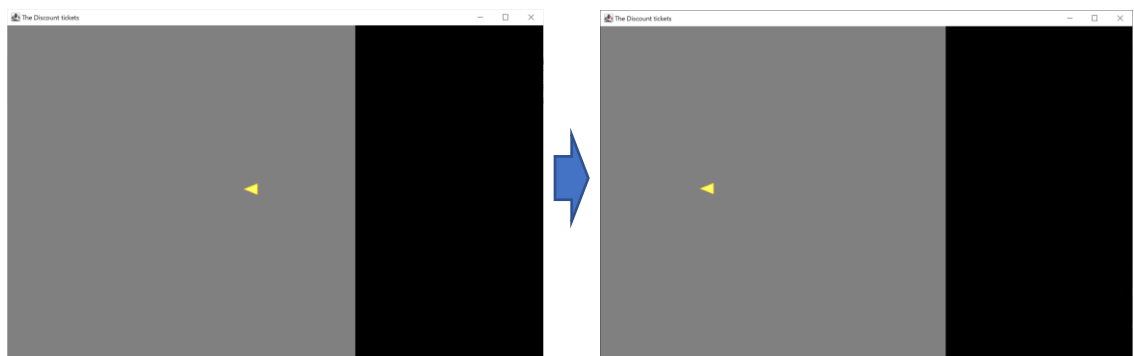
・ 下向き移動



・ 右向き移動



・ 左向き移動



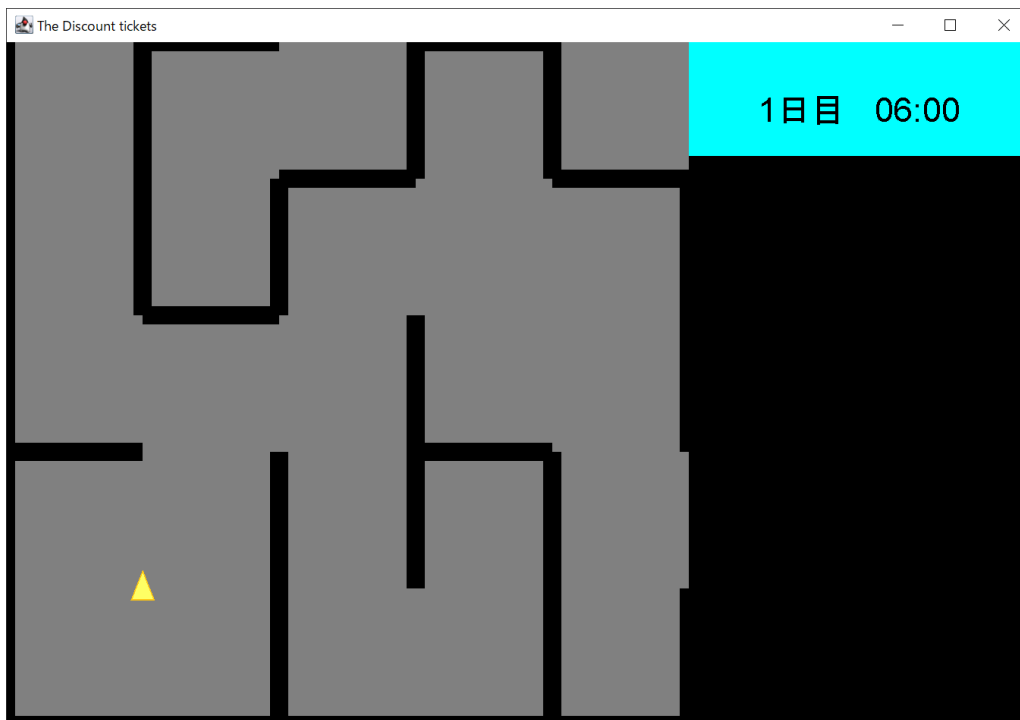
—SP2—

<SP 2 要件定義>

- SP1 からの微調整
 - ・ウィンドウの描画エリアを 900×600 とし、全体のサイズは描画エリアに淵やタイトルバーを含めた大きさへ変更
 - ・迷路画面のサイズを 600×600 へ変更
 - ・クーポン画面を 600×300 へ変更
- 迷路画面に壁を配置される
 - ・表示される壁の配置は 9 種類 (Maze1~9)
 - ・最初は Maze7
- プレイヤーは壁をすり抜けられない
- プレイヤーが迷路画面外に出ると隣の迷路へ移動する
- クーポン画面の上部に現在時刻が表示される

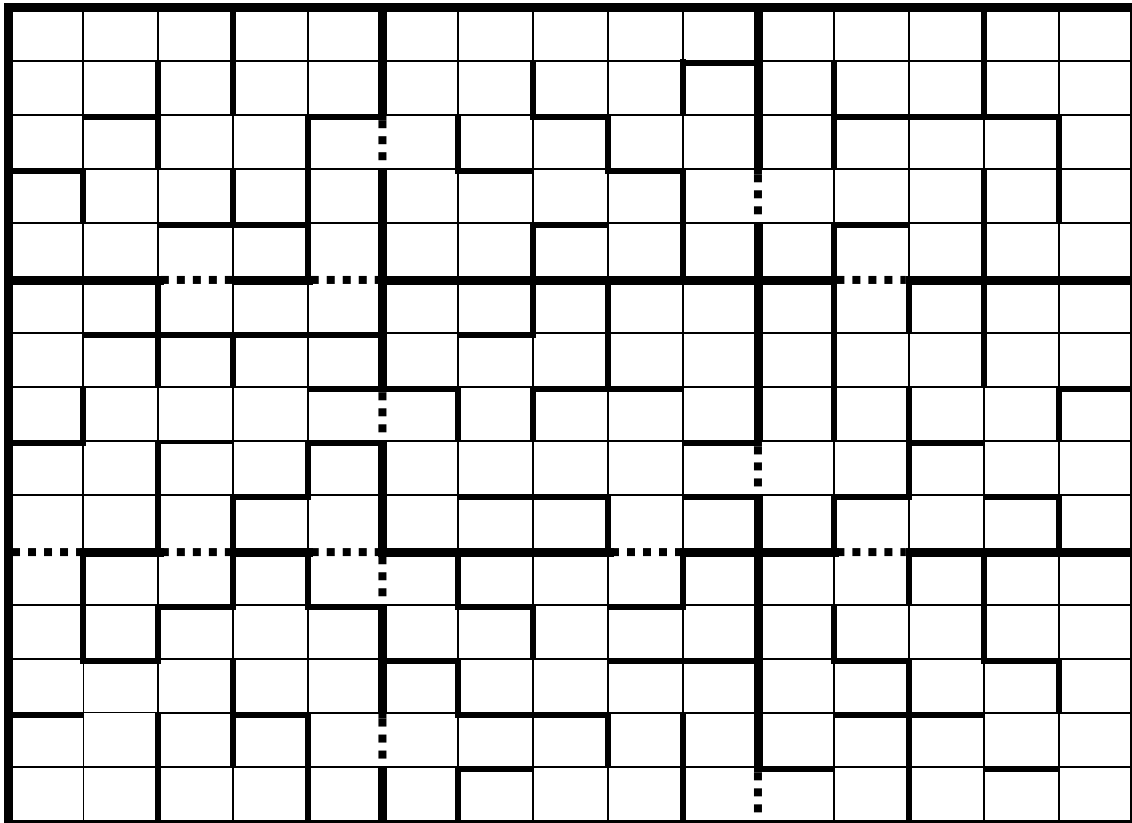
<SP 2 外部設計>

- ゲーム開始時のウィンドウ



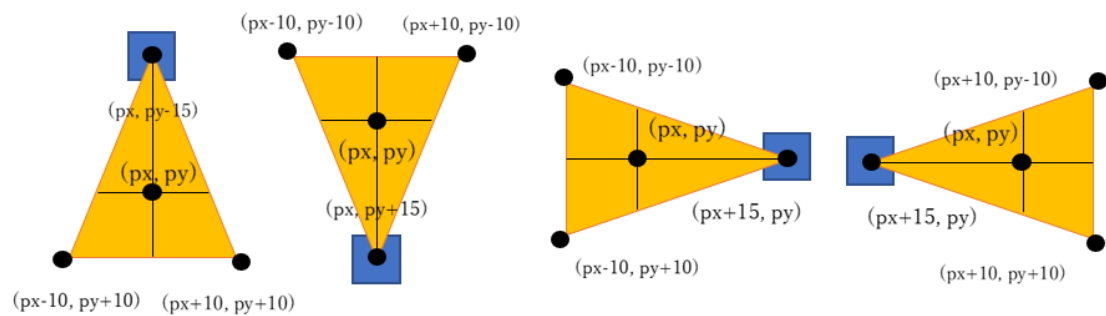
●迷路全体図

- ・各迷路の番号は，電話のダイヤル配列の順に 1~9 が割り振られている．
- ・太線は壁，破線は各迷路の出入り口



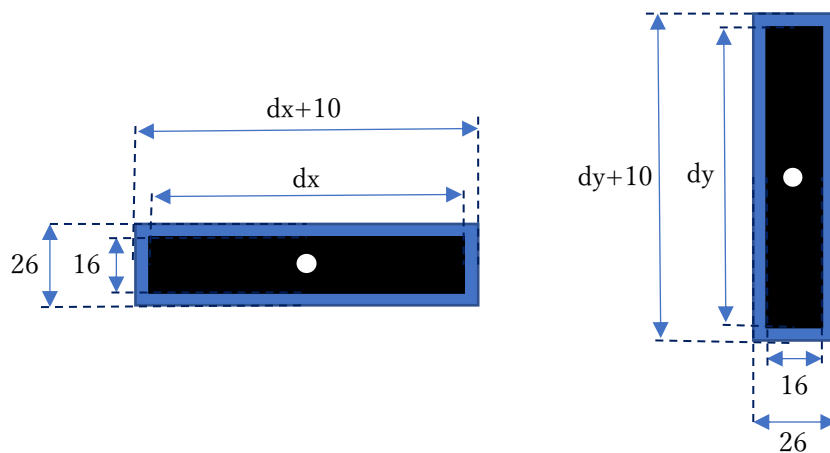
●プレイヤーの衝突判定範囲

- ・青い四角形が判定を行う範囲
- ・判定範囲は一辺 5 の正方形



●壁とその衝突判定範囲

- ・ 壁の厚さは 16
- ・ 白点が壁の位置を表す座標であり，その壁の重心
- ・ 青い長方形が判定を行う範囲
- ・ 判定範囲は重心を共有する一辺 10 だけ大きい長方形



<SP 2 内部設計>

●18 個のクラス

●GameManager

- ・ 一定時間ごとに現在時刻を更新する
- ・ 一定時間ごとにプレイヤーが迷路外へ出ていないかの判定
- ・ TicketsPanel を再描画する

●Player

- ・ GameChar の派生クラス
- ・ キーイベントが起こると壁との接触判定を行う
- ・ 向いている方向に合わせて，衝突判定範囲の中心座標を更新
- ・ 現在の迷路を出たとき，新たな座標を得る

●TicketsPanel

- ・ 再描画時に，GameManager と TimePanel を描画する

●TimePanel

- ・ JPanel
- ・ 現在時刻を表示
- ・ TicketsPanel 上に存在する
- ・ TicketsPanel によって再描画される

●GameChar

- ・ ゲームオブジェクトの基底クラス

●Wall

- ・ GameChar の派生クラス
- ・ 壁の生成と描画を行う

●Maze

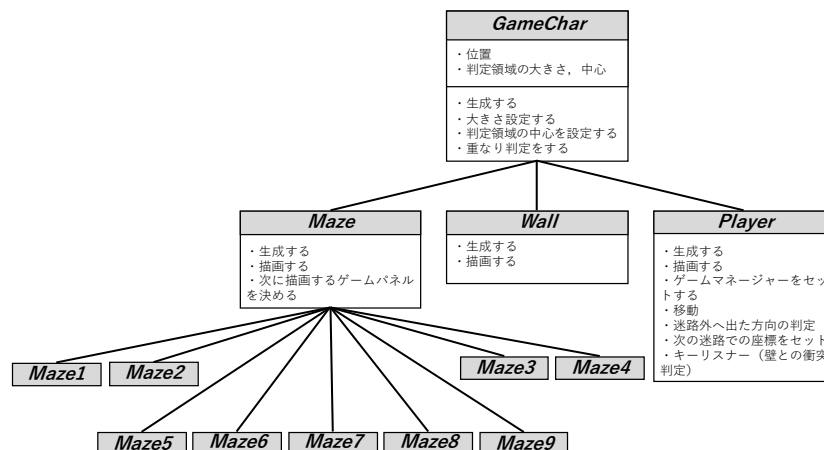
- ・ GameChar の派生クラス
- ・ Maze1~9 の基底クラス

●Maze1~9

- ・ Maze の派生クラス
- ・ 壁の設置を行う
- ・ 隣に位置する Maze を記憶

●クラスの階層

- ・ 迷路パネル上に登場する全てのゲームオブジェクトの基底クラス GameChar
- ・ Maze 1 ~ 9 の基底クラス Maze
- ・ Player、Wall も GameChar の派生クラス



●Maze1~9 が MazePanel 上に生成されると同時に, GameManager が持つ ArrayList<Wall> walls も更新

●プレイヤーが迷路外へ出ていないかの判定で行う処理

- ・ Player クラスがもつ, どの方向の外に出たかを判定する out() を実行. Out() は外に出た方向に応じて int 型を返す. (上: 0, 下: 1, 右: 2, 左: 3, 出ていない: 4)
- ・ プレーヤーが外へ出ていれば, 出た方向に応じた迷路に更新
- ・ プレーヤーの座標を隣の迷路画面における座標へ変換し, プレーヤーがもつ GameManager を更新

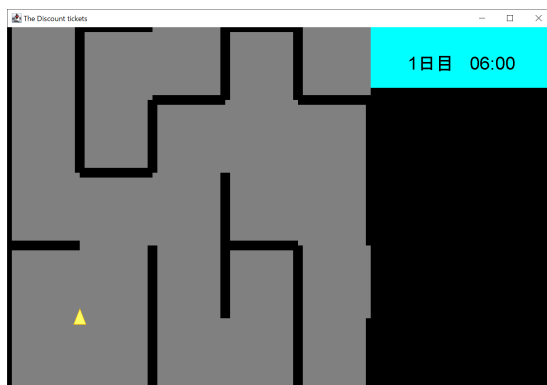
●キーイベントによるプレイヤーと壁の衝突判定

- ・ 現在の方向と押された矢印キーの方向が違えば方向のみを更新する.
- ・ 現在の方向と押された矢印キーの方向が同じであれば移動量を更新する. その後現在表示されているすべての壁との接触判定を行い, いつでも接触している壁があれば移動量を 0 に再更新する.

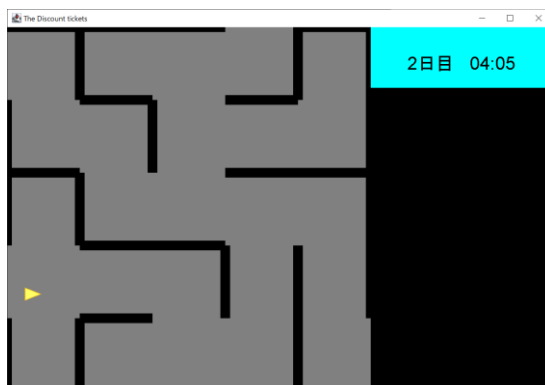
●TicketsPanel を再描画するたびに GameManager が持つ現在時刻を文字列へ変換して TimePanel 上に描画する.

<SP 2 の進捗>

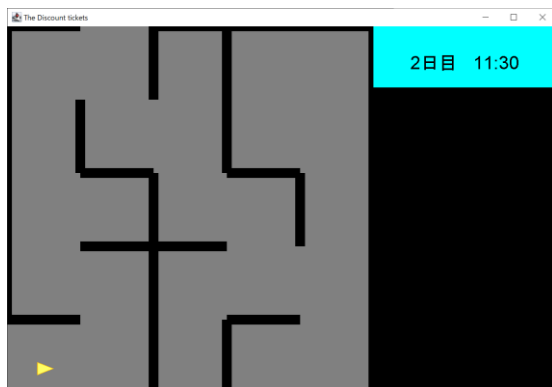
・ Maze7(スタート地点)



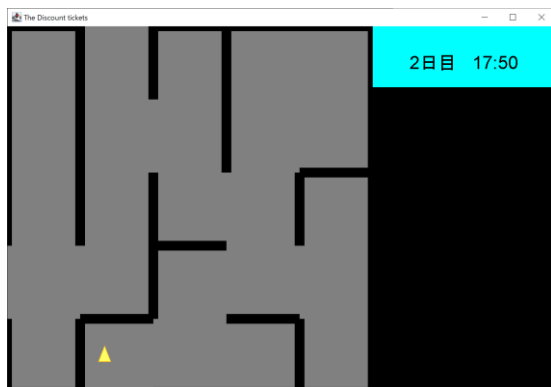
・ Maze8(Maze7 から Maze8 へ)



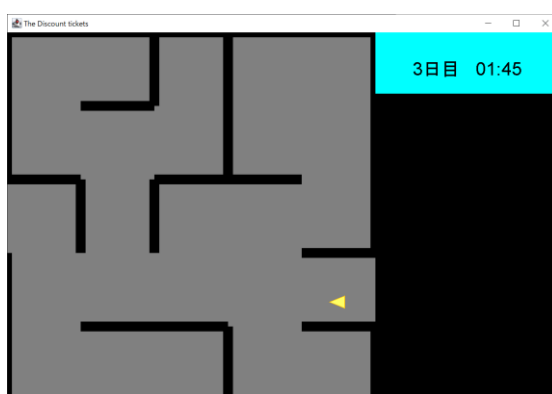
・ Maze9(Maze8 から Maze9 へ)



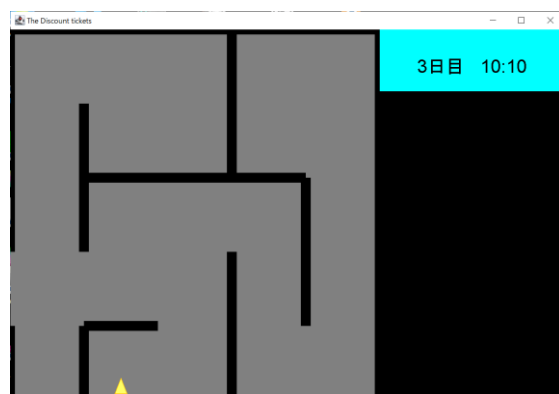
・ Maze6(Maze9 から Maze6 へ)



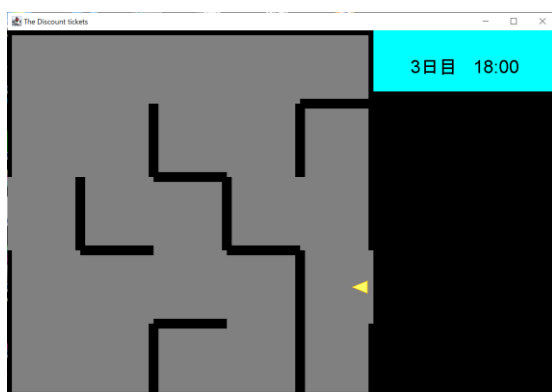
・ Maze5(Maze6 から Maze5 へ)



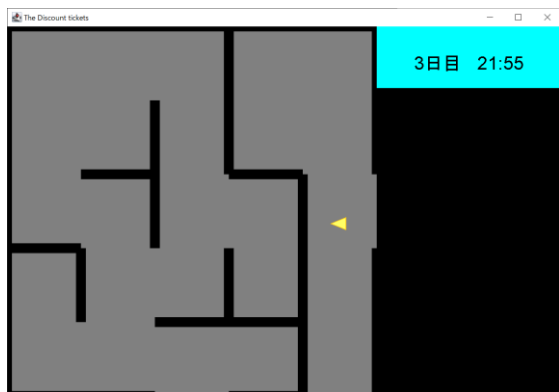
・ Maze3(Maze6 から Maze3 へ)



・ Maze2(Maze3 から Maze2 へ)



・ Maze1(Maze2 から Maze1 へ)



・ Maze4(Maze1 から Maze4 へ)



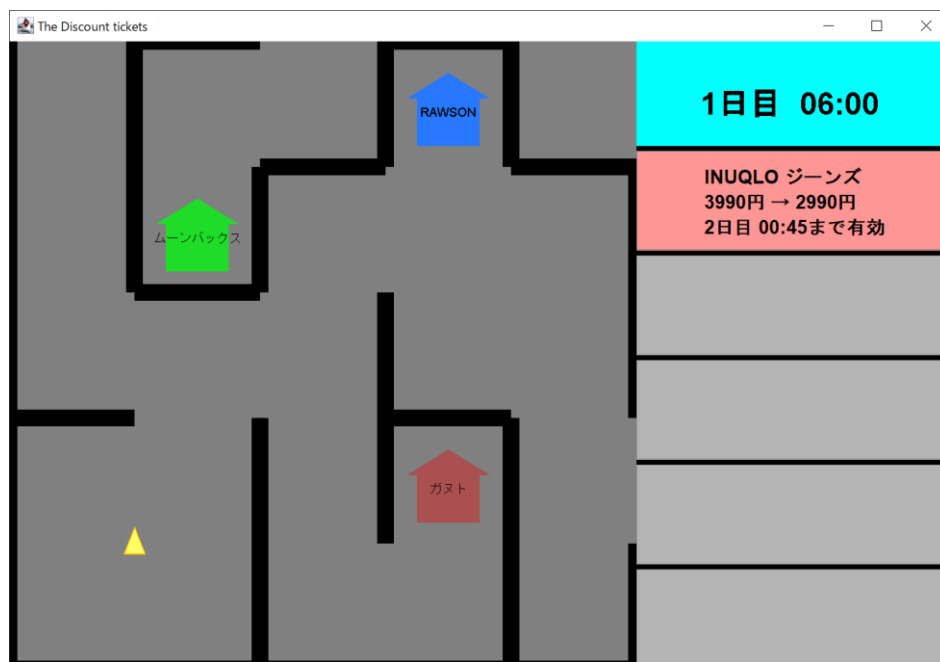
—SP3—

<SP 3 要件定義>

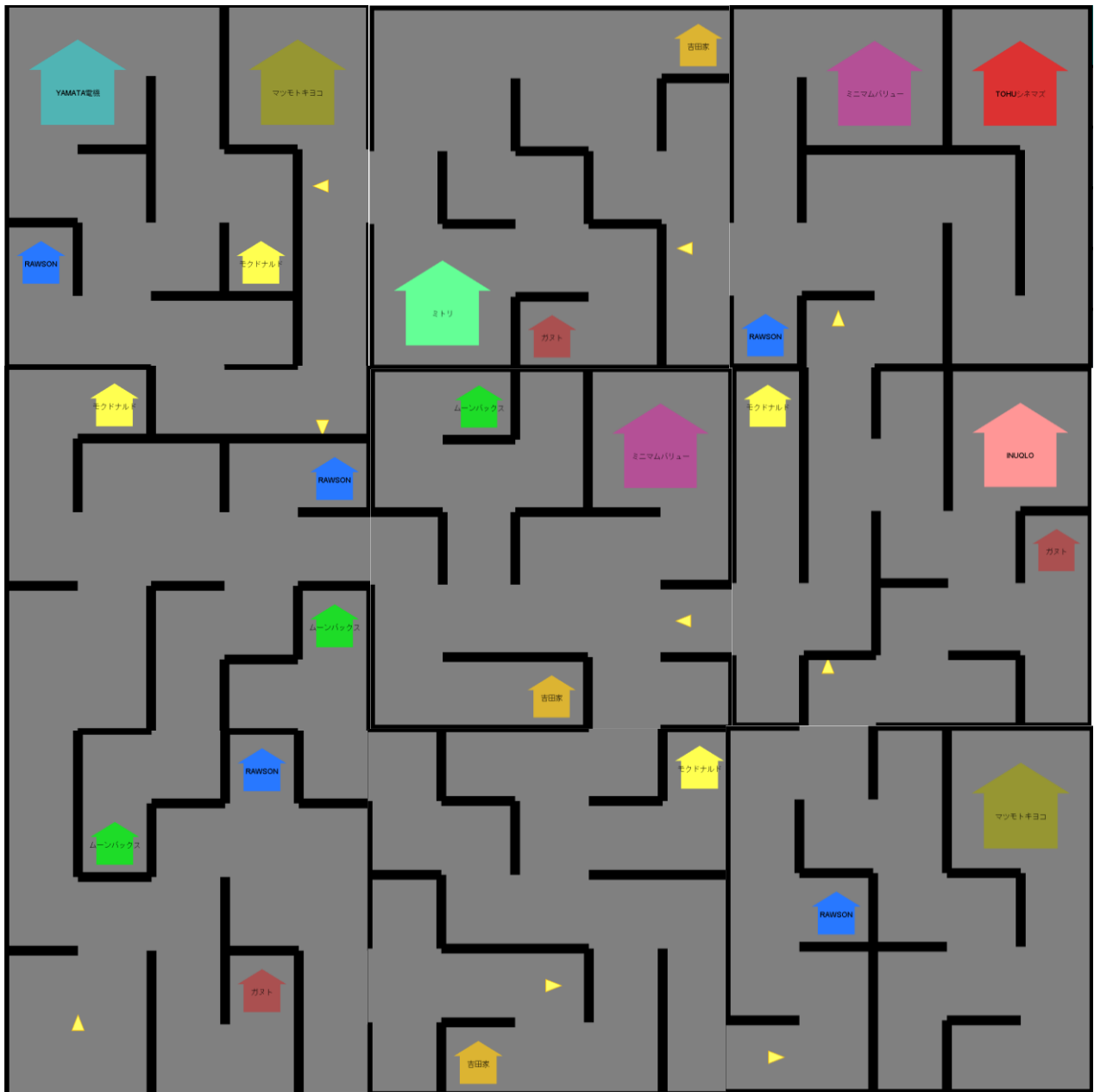
- 迷路画面に店が配置される
 - ・表示される店は 11 種類
- クーポン画面の時刻表示欄の下部にクーポン券が JButton として配置される.
 - ・クリックしたら消える
 - ・有効期限が過ぎたら消える

<SP 3 外部設計>

- ゲーム開始時のウィンドウの例



●迷路全体図



<SP 3 内部設計>

●21 個のクラス

●GameManager

- ・一定時間ごとに現在時刻を更新するとともに、クーポン券の残りの有効期限を減らす
- ・一定時間ごとにクーポンを生成

●TicketsPanel

- ・再描画時に, CouponPanel と TimePanel を描画する

●CouponPanel

- ・生成時に 5 つのデフォルトの JButton を, 配列を用いて設置
- ・アクションコマンドを配列の番号にセット

●Maze1~9

- ・店が設置される

●Shop

- ・GameChar クラスを継承
- ・店を生成する
- ・店によって色や大きさが異なる

●Coupon

- ・クーポン券 (内部情報) を生成する.
- ・内部情報として, 定価, 割引価格, 有効期限, int 型の使用可能店舗, String 型の使用可能店舗名, 商品名, クーポン券に表示するテキストを所持する.
- ・生成されるクーポン券はランダムに決定

●デフォルトボタン

- ・フォント (色: 黒, フォント名: DIALOG, スタイル: BOLD, サイズ 18)
- ・テキストは無し
- ・背景色 Color(180, 180, 180)
- ・サイズ (300, 95)
- ・Enable false

●クーポン券ボタン

- ・フォント (デフォルトボタンのまま)
- ・テキスト ("- ・背景色 店の色と同じ
- ・サイズ (デフォルトボタンのまま)
- ・Enable true

●make_Coupon_Button()

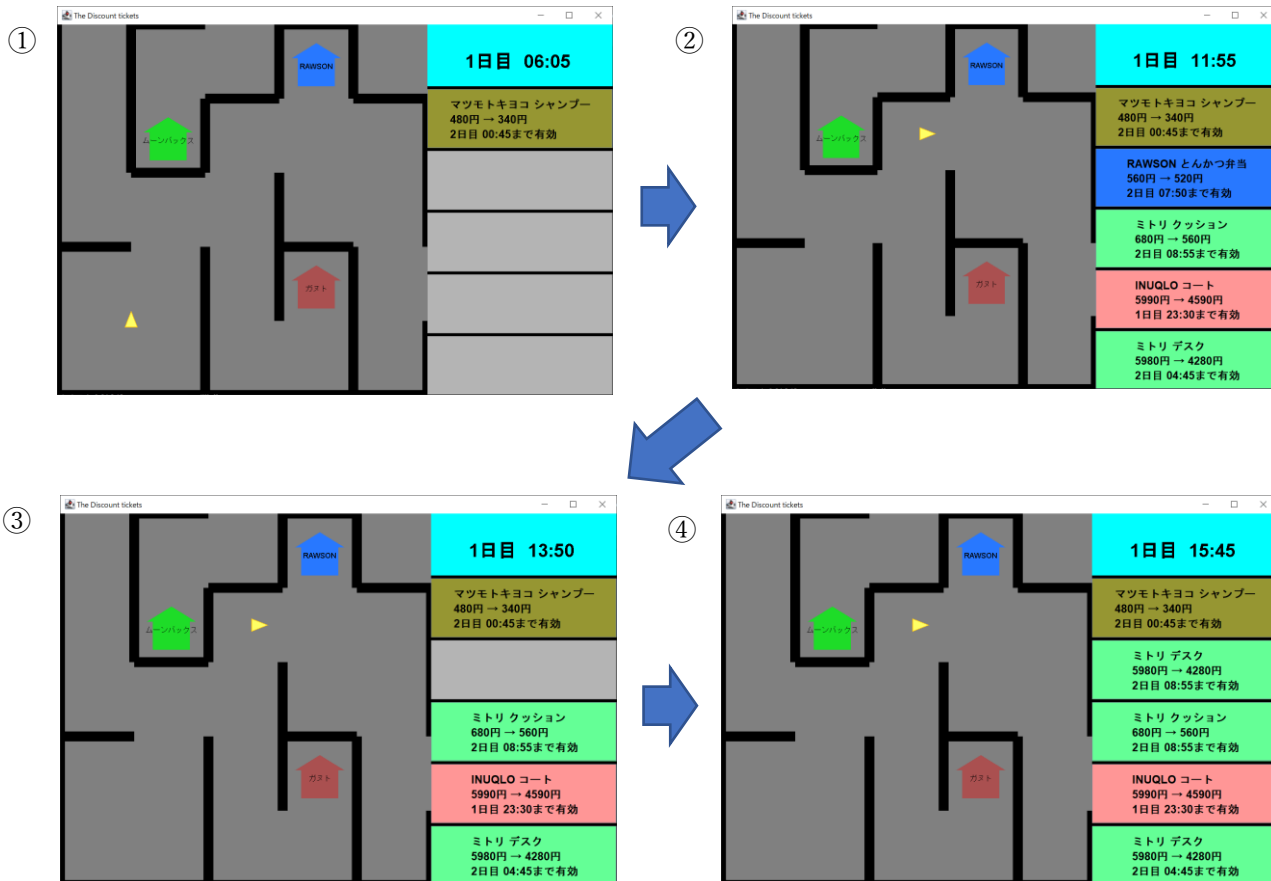
- ・ ボタンの配列の中からデフォルトボタン JButton button[i]を探す
- ・ 新たなクーポンを生成し、Coupon 型 coupon[i]に代入
- ・ coupon[i]の所持する情報から、button[i]をクーポン券ボタンに変更
- ・ 一度クーポン券ボタンを作ったら、それ以降の配列内にデフォルトボタンがあっても処理は行わない。

●アクションイベントによるクーポン券の利用

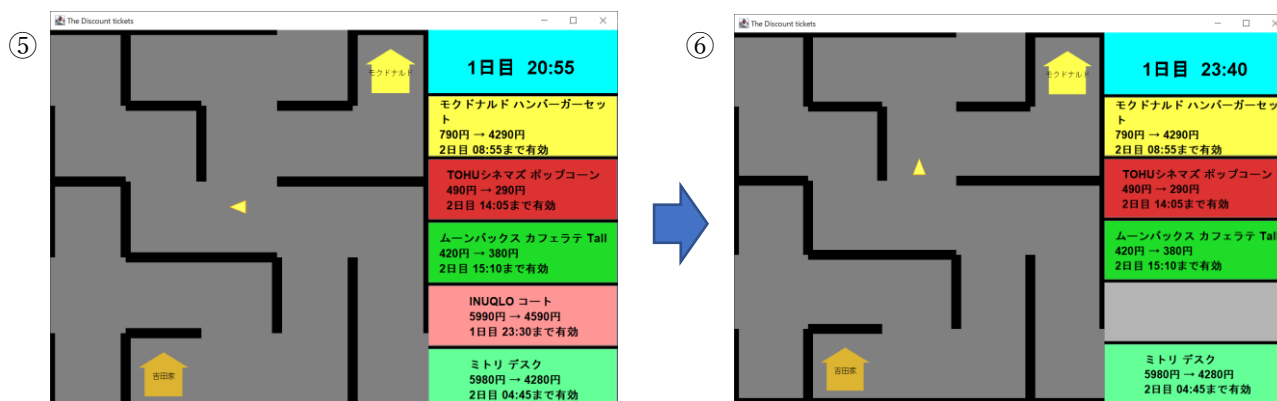
- ・ actionPerformed();メソッドを用いて押されたボタンの配列の番号を String 型で取得する
- ・ 取得した番号を int 型へ変換する
- ・ 取得した番号に対応するボタンをデフォルトに戻す。
- ・ 取得した番号に対応するボタンより配列の後ろにあるボタンのフォーカスを false にする。

<SP 3 の進捗>

起動後①のようなウィンドウが表示された。しばらくたつとクーポン券が 1 つずつ増え

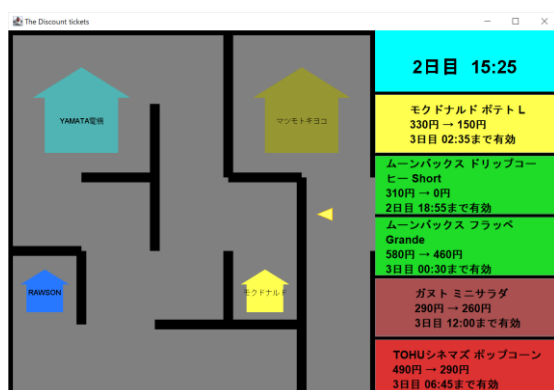


ていき、最終的には②のように5つのボタンすべてがクーポン券になった。その後、上から2番目のクーポン券をクリックすると③のようにクーポン券は消え、しばらくすると④のように新たにクーポン券ボタンが生成された。また⑤から⑥のように、有効期限が切れたクーポン券は自然に消えた。

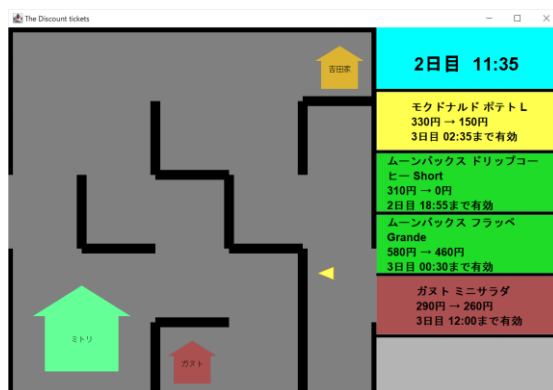


また、各 Maze は以下ようになった。

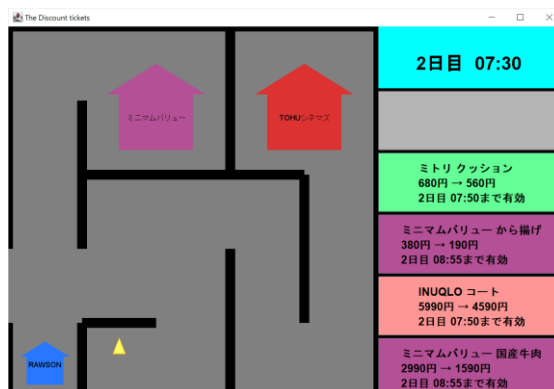
・ Maze_1



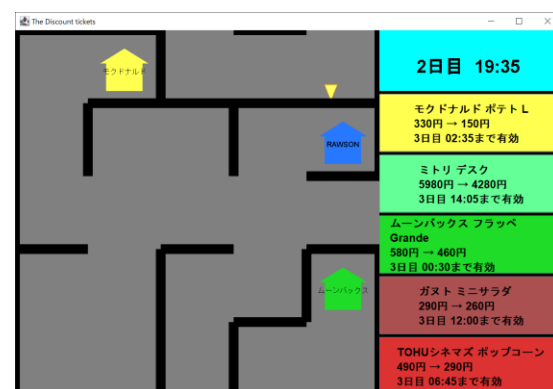
・ Maze_2



・ Maze_3



・ Maze_4



・ Maze_5

2日目 04:55	
ミニマムバリュー から揚げ	380円 → 190円 2日目 07:00まで有効
ミトリ クッション	680円 → 560円 2日目 07:50まで有効
ミニマムバリュー から揚げ	380円 → 190円 2日目 08:55まで有効
INUQLO コート	5990円 → 4590円 2日目 07:50まで有効
ミニマムバリュー 国産牛肉	2990円 → 1590円 2日目 08:55まで有効

・ Maze_6

1日目 22:35	
ミニマムバリュー から揚げ	380円 → 190円 2日目 07:00まで有効
ミトリ クッション	680円 → 560円 2日目 07:50まで有効
ミニマムバリュー から揚げ	380円 → 190円 2日目 08:55まで有効
INUQLO コート	5990円 → 4590円 2日目 07:50まで有効
ミニマムバリュー 国産牛肉	2990円 → 1590円 2日目 08:55まで有効

・ Maze_7

1日目 06:25	
ミニマムバリュー から揚げ	380円 → 190円 2日目 07:00まで有効

・ Maze_8

1日目 12:35	
ミニマムバリュー から揚げ	380円 → 190円 2日目 07:00まで有効
ミトリ クッション	680円 → 560円 2日目 07:50まで有効
ミニマムバリュー から揚げ	380円 → 190円 2日目 08:55まで有効
TOHUシネマズ ポップコーン	490円 → 290円 2日目 03:40まで有効

・ Maze_9

1日目 16:55	
ミニマムバリュー から揚げ	380円 → 190円 2日目 07:00まで有効
ミトリ クッション	680円 → 560円 2日目 07:50まで有効
ミニマムバリュー から揚げ	380円 → 190円 2日目 08:55まで有効

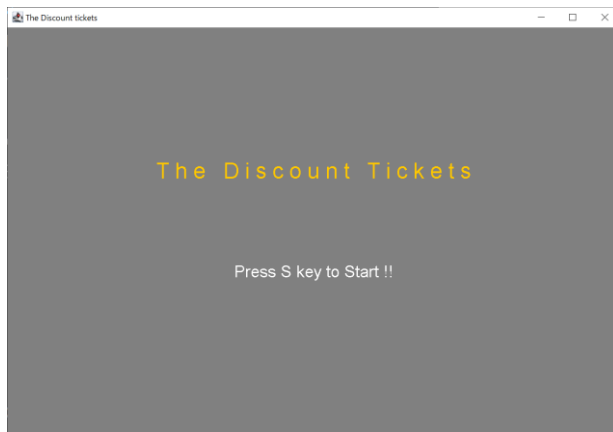
—SP4—

<最終（SP4）要件定義>

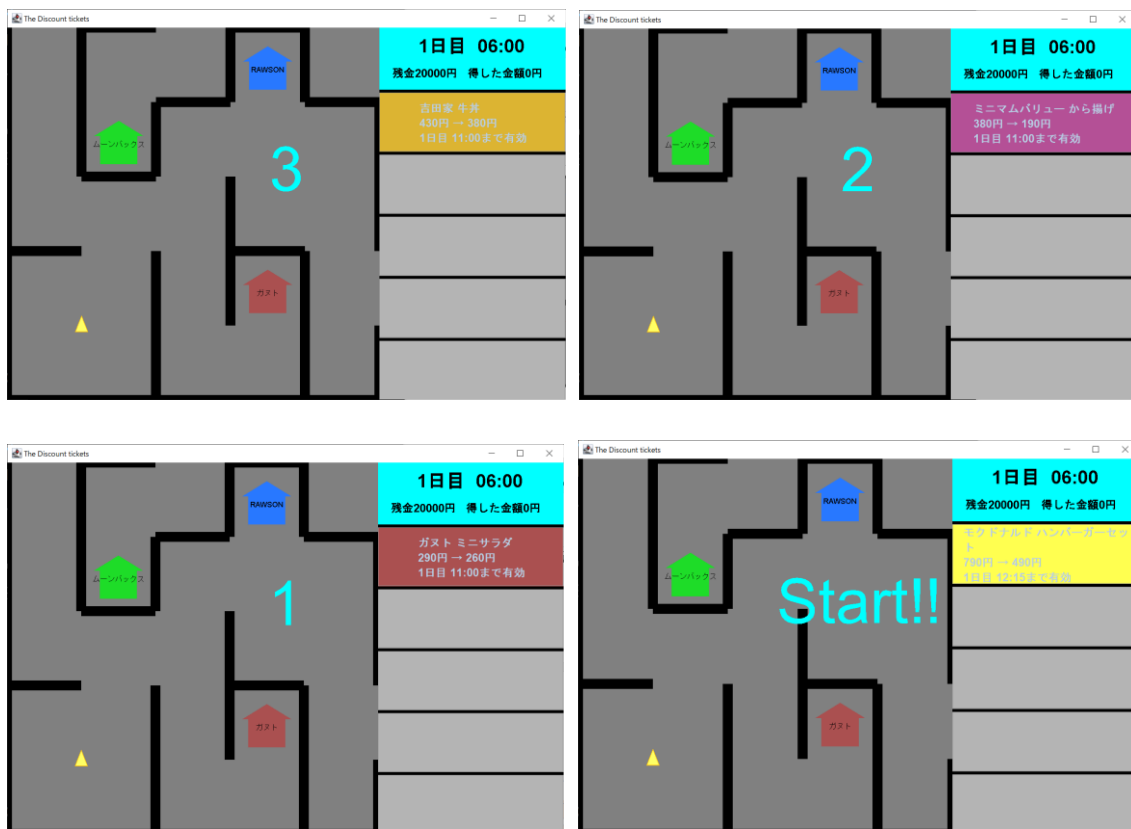
- 現在時刻の下に残金と得した金額を表示
- プレイヤーが店と接触した状態でその店のクーポンをクリックするとクーポン券を使う
 - ・クーポン券に応じて残金が減り，得した金額が増える
 - ・クーポン券に応じて時間が経過する
- 表示画面の切り替え
 - ・スタート画面でSキーを押すとゲームスタート
 - ・ゲーム画面で制限時間が経過，もしくは残金がゼロになるとスタート画面に戻る
 - ・ゲームプレイ後のスタート画面では，ゲーム名の代わりにスコア（得した金額）が表示される

<最終（SP4）外部設計>

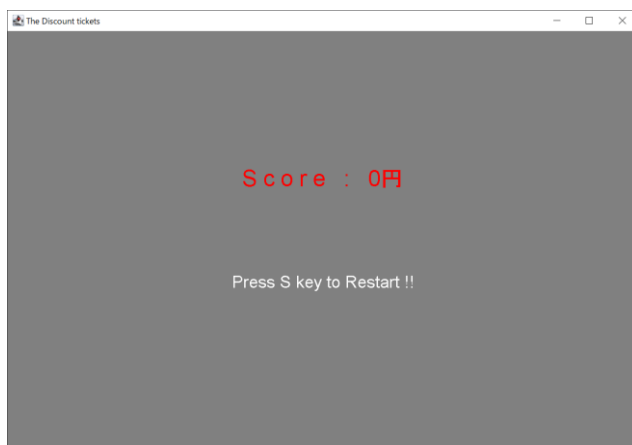
- スタート画面



●ゲーム画面



●スタート画面（ゲームプレイ後）



<最終（SP4）内部設計>

●27 個のクラス

●GameManager

- ・ ステージが StartStage ならば, mazePanel と ticketsPanel を見えなくする.
- ・ ステージが StartStage か GameStage かによってループ処理を変える.

●BackPanel

- ・ 背景のパネル
- ・ mazePanel と ticketsPanel が貼り付けられている

●Msg

- ・ メッセージの基底クラス

●MsgBrink

- ・ 点減するメッセージのクラス
- ・ Msg の派生クラス

●Stage

- ・ ステージの基底クラス

●StageStart

- ・ スタート画面
- ・ Stage の派生クラス
- ・ ゲーム名の「The Discount Tickets」と指示文「Press S key to Start !!」が表示される
- ・ ゲームプレイ後であれば, スコアと指示文「Press S key to Restart !!」が表示される
- ・ S キーを押すとゲーム画面に切り替わる

●StageGame

- ・ ゲーム画面
- ・ Stage の派生クラス
- ・ スタート画面から移ってくると, 「3」, 「2」, 「1」, 「Start!!」とカウントダウンが表示される.
- ・ カウントダウン終了後, ゲームが始まる
- ・ 制限時間が経過するか残金がなくなると, スタート画面に切り替わる

●Coupon

- ・ クーポンを使える店の営業時間を記憶

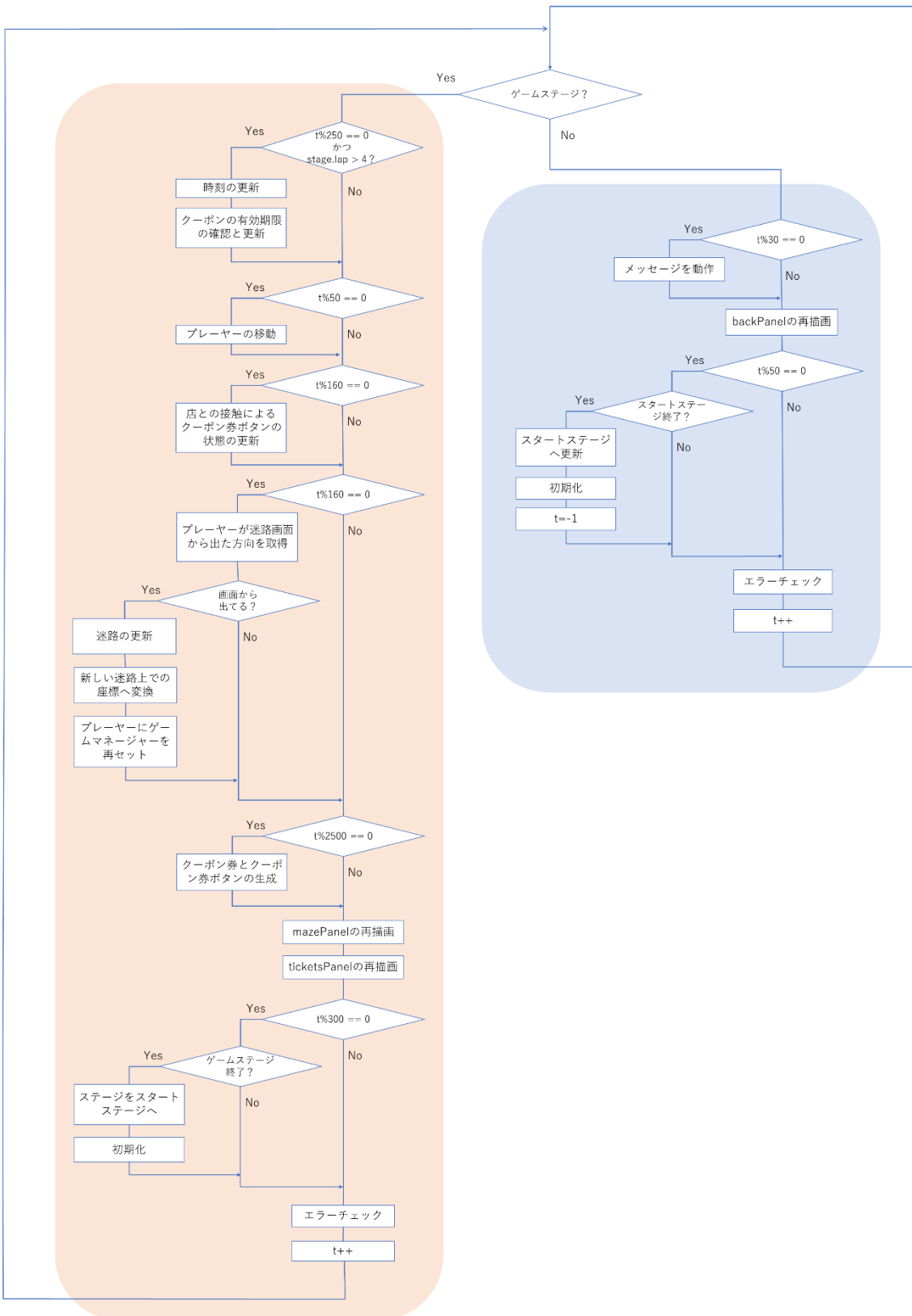
●CouponPanel

- ・ 接触している店が営業時間内であれば、その店のクーポンボタンを `setEnabled(true)` に
- ・ クーポンボタンがクリックされたら、残金、得した金額、現在時刻を更新し、デフォルトボタンに戻す

●TimePanel

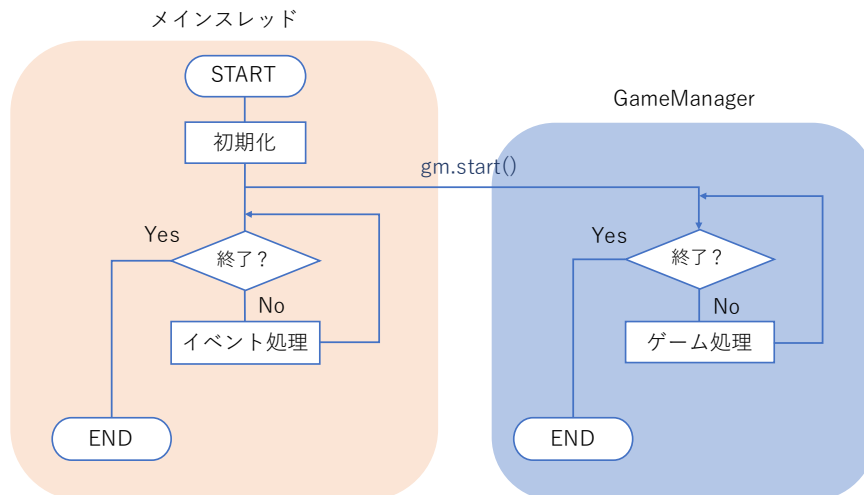
- ・ 現在時刻の下に残金と得した金額を表示

●ゲームマネージャー内のフローチャート



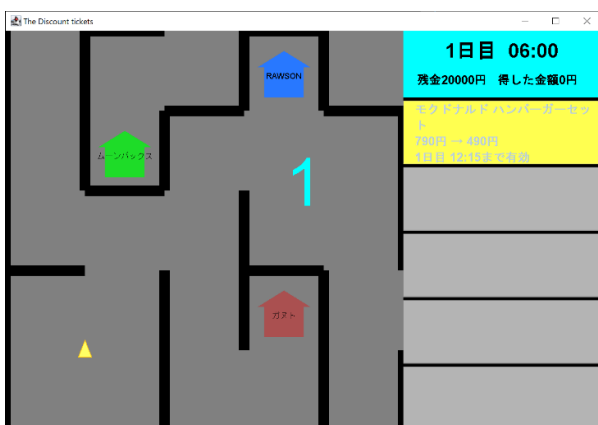
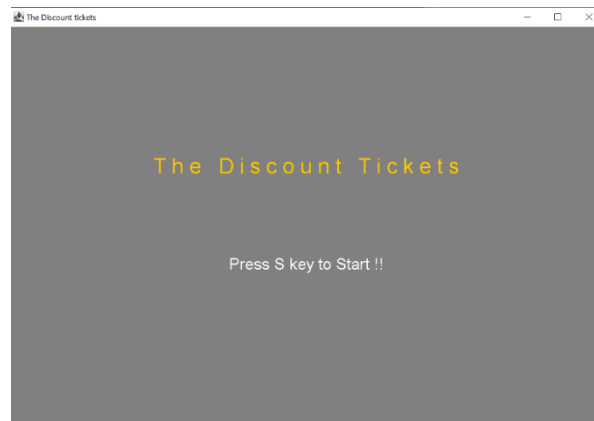
●全体のフローチャート

青で示された GameManager の処理が、上記で示したゲームマネージャー内のフローチャートに該当する。そのため、Game マネージャースレッドで終了することはなく、無限にループを繰り返す。



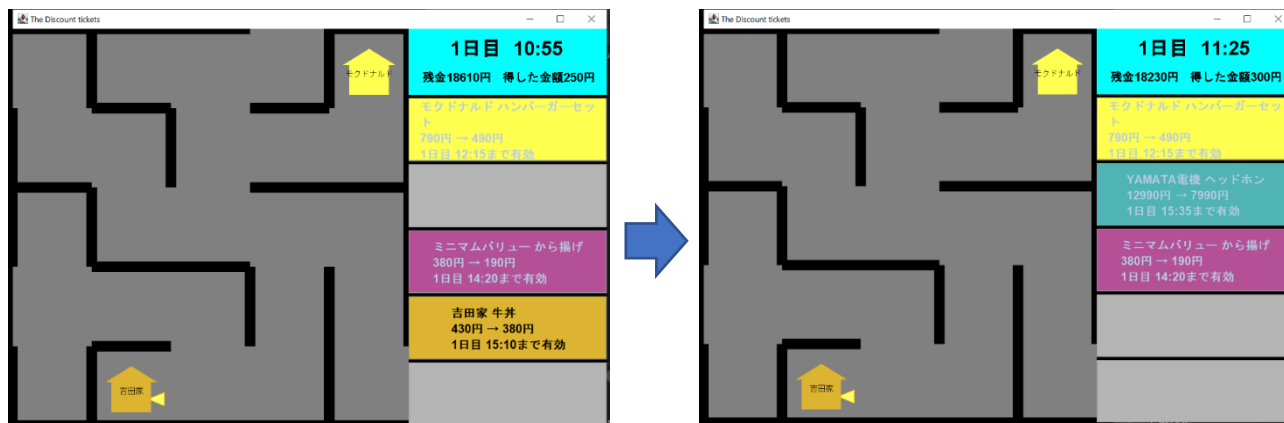
<最終 (SP4) 進捗>

ゲームウィンドウを開くとスタート画面が表示され、S キーを押すとカウントダウンが始



まってゲームが開始した。

また、プレーヤーが店と接触すると、その店のクーポンの文字が黒色に変わった。この状態でクーポンをクリックするとそのクーポンは消え、支払った額だけ残金が減り、得した金額が増加した。



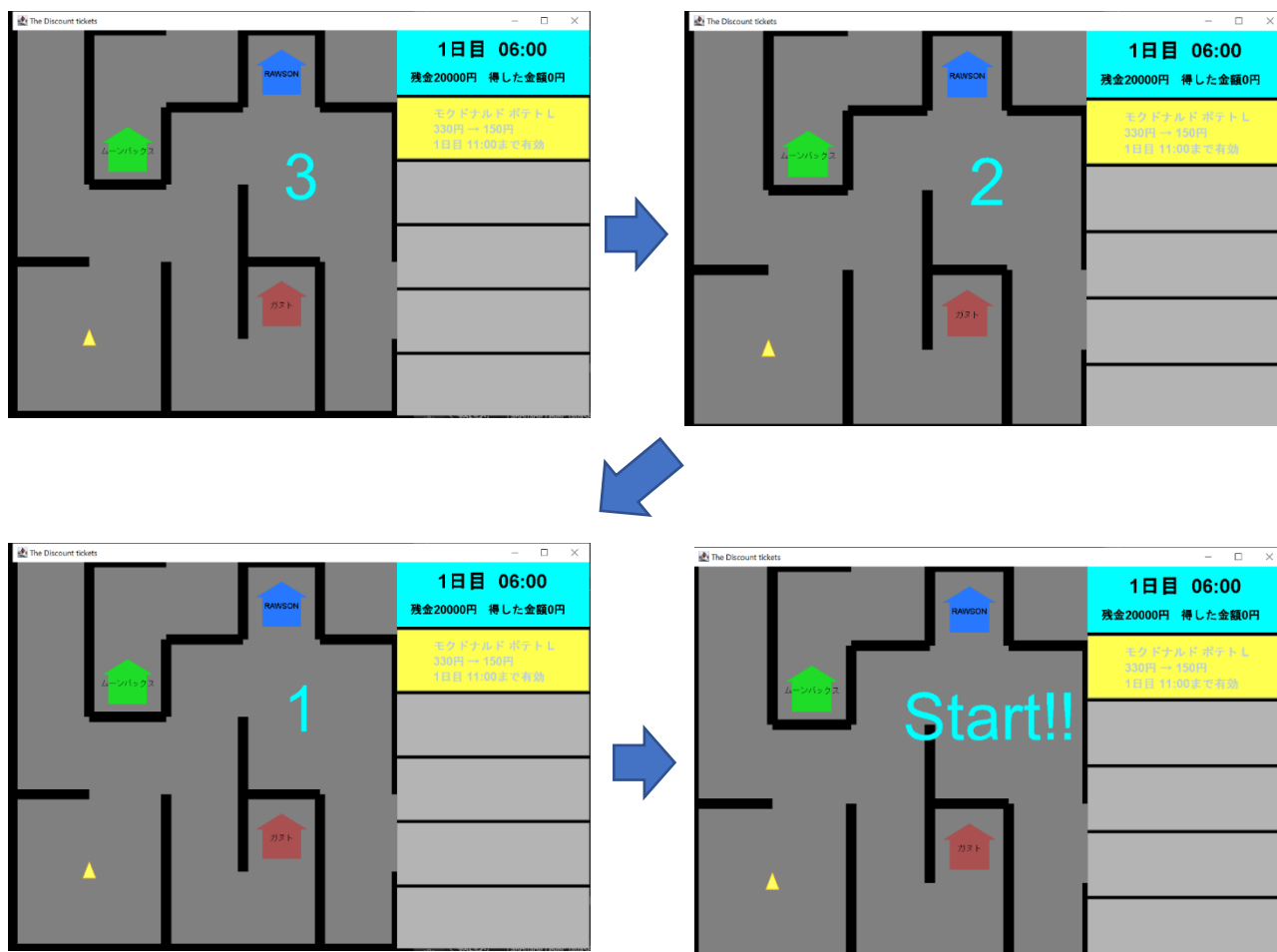
さらに該当するクーポンが複数ある場合は、その全てのクーポンが使用可能になった。また、有効期限が過ぎたクーポンは自動的に削除された。



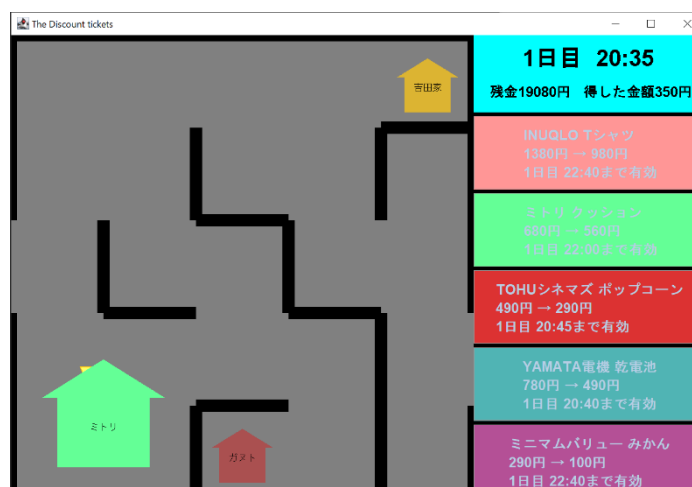
4 日目の 6:00 になるとスタート画面に戻り、ゲーム名の代わりにスコアが表示された。



S キーを押すとカウントダウン始まり、ゲームが再スタートした。



また、営業時間外の店と接触しても、その店のクーポンは使用可能な状態にはならなかった。
(ミトリの営業時間は 11:00~20:00)



残金より高額な買い物をするとゲームは終了し、スタート画面に切り替わった。



～参考資料～

- ・【簡単理解】Java での乱数の使い方まとめ エンジニアの入り口
<<https://eng-entrance.com/java-math-random>>
- ・BoxLayout クラス Let's プログラミング
<<https://www.javadrive.jp/tutorial/boxlayout/>>
- ・インセットの値を取得する Let's プログラミング
<<https://www.javadrive.jp/tutorial/insets/index3.html>>
- ・ボタンに表示される文字列のフォントを設定する Let's プログラミング
<<https://www.javadrive.jp/tutorial/jbutton/index4.html>>
- ・ボタンの前景色/背景色の設定と背景の透明/非透明の切り替え Let's プログラミング
<<https://www.javadrive.jp/tutorial/jbutton/index2.html>>
- ・ボタンのサイズをピクセル単位で設定する Let's プログラミング
<<https://www.javadrive.jp/tutorial/jbutton/index5.html>>
- ・ボタンに表示される文字列の水平及び垂直位置を設定する Let's プログラミング
<<https://www.javadrive.jp/tutorial/jbutton/index7.html>>
- ・ボタンの有効/無効を切り替える Let's プログラミング
<<https://www.javadrive.jp/tutorial/jbutton/index14.html>>
- ・ActionEvent Let's プログラミング
<<https://www.javadrive.jp/tutorial/event/index2.html>>
- ・Swing によるクライアントデザイン② 【サラリーマン応援サイト】Japan IT エンジニア
<<http://japanengineers.seesaa.net/article/383406375.html>>
- ・GUI 部品にフォーカスを奪われた場合の対処: Java Tips ～Java プログラミング編 @IT
<<https://atmarkit.itmedia.co.jp/ait/articles/0501/19/news131.html>>
- ・【Java】キー入力処理 のんぼぐ軟弱クリエイターの備忘録
<<https://nompor.com/2017/12/10/post-1924/>>
- ・中間コンテナ Java 入門
<<http://msugai.fc2web.com/java/Swing/intermediate.html>>
- ・【Java】JFrame のサイズにはタイトルバーや枠線も含まれている! ゆーじの Unity 開発日記
<<https://unity-yuji.xyz/java-jframe-size-insets/>>
- ・ウィンドウを表示する前にタイトルバーの高さを取得する方法 ゆーじの Unity 開発日記
<<https://unity-yuji.xyz/java-get-title-bar-before-show-window/>>
- ・Java で使えるフォント名 MLT Lab のプログラミングブログ
<<https://www.mltlab.com/wp/archives/67>>