仕様書

間違い探しアプリ作成チーム　「チャー班」

2012/09/28

テスト用に用意する機種

Android2.2.2搭載スマートフォン　512MB　1GHz

Android4.0.3搭載スマートフォン　1GB 1.5GHz

使用する画像は主に岩手県内の風景など

候補地一覧

・皿ブレッド

・わんこそば

・冷麺

・じゃじゃ麺

・ETロボコン

・

・学校（各教室、外観）

・北上川（海運橋）

・岩手公園

・鬼の手

・岩手山

・盛岡駅

・川口（わ～だ～）

・温泉

・宮古、浄土ヶ浜

・マリオス

・田んぼ

・花巻駅

・鈴木

・新幹線

・八幡平（古代の息吹）

・小岩井周辺（一本桜、USHI）

・県庁（石割桜）

・平泉

・御所湖

・仙人峠

・樹海ライン

・県営球場

・アスピーテライン

・地獄谷

・焼走

・大通り

・加茂君のパソコンのダウンロードフォルダ　△▼△▼△▼△▼重要▼△▼△▼△▼△▼

・岩泉（龍泉洞）

・沿岸（北山崎）

・飯岡駐在所（阿部君）

・一関（猊鼻渓、厳美渓）

・花巻駅の壁画（古代文明）

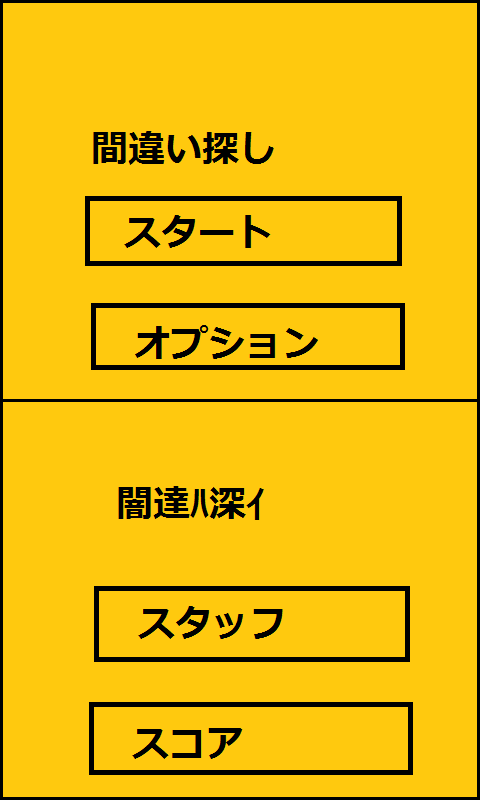
・サファリパーク

・岩山動物園

開発環境 Eclipseを使用

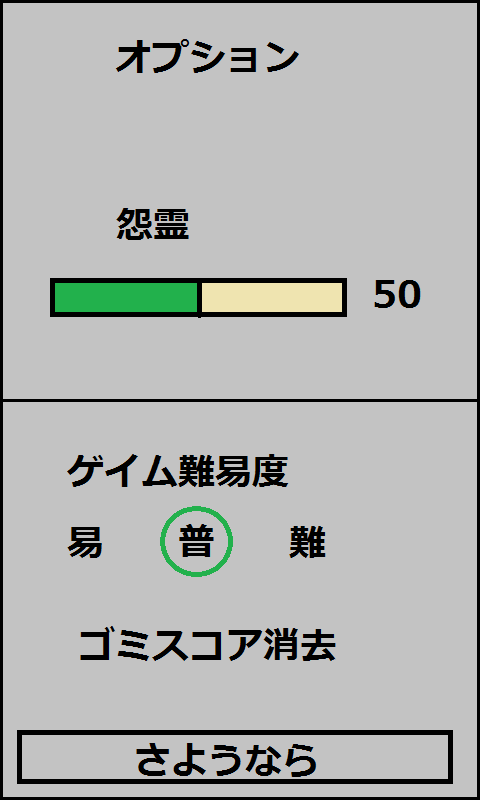
notepad.exeは使用しない

1. 主に使用する画面
   1. メニュー画面
   2. オプション画面
   3. クレジット画面
   4. スコア画面
   5. プレイ画面
   6. ゲイムオーバー画面
2. 使用する変数
   1. int level
3. メニュー画面
   1. xmlファイル設計　menu.xml
      1. 画面中央で水平方向に分割
      2. 画面は縦に使用する
      3. TitleTextField 上部にタイトルロゴ用のテキストフィールド
      4. PlayButton 上部にプレイ用ボタン
      5. OptionButton 上部にオプション用ボタン
      6. FakeTitleTextField 下部に偽タイトルロゴ用のテキストフィールド
      7. StaffButton 下部にクレジット用ボタン
      8. ScoreButton 下部にスコア用ボタン



* 1. Javaファイル設計　menu.java
     1. xml各オブジェクトの関連付け
     2. プレイ画面への遷移イベント
     3. オプション画面への遷移イベント
     4. クレジット画面への遷移イベント
     5. スコア画面への遷移イベント
     6. マスクデータとして難易度変数private int level保持
        1. 難易度変数 private int level は 0 易 1 普通 2 難
  2. その他ファイル
     1. Title.png タイトルロゴ画像
     2. FakeTitle.png 偽タイトルロゴ画像
     3. Play.png プレイボタン画像
     4. Option.png オプションボタン画像
     5. Staff.png クレジットボタン画像
     6. Score.png スコアボタン画像
     7. MenuBack.png 背景画像

1. オプション画面
   1. xmlファイル設計　option.xml
      1. 画面中央で水平方向に分割
      2. 画面は縦に使用する
      3. OptionTitleTextField 上部にオプションタイトルのテキストフィールド
      4. VolumeTextField 上部に音量のテキストフィールド
      5. SoundVolumeTextField の右に現在の音量のパーセンテージを表示するテキストフィールド
      6. LevelTextField 下部に難易度のテキストフィールド
      7. EasyButton 難易度のテキストフィールド左下に易ボタン
      8. NomalButton 易ボタンの右に普通ボタン
      9. HardButton 普通ボタンの右に難ボタン
      10. ScoreEraseButton 下部にスコア消去ボタン
      11. ReturnButton 下部に戻るボタン



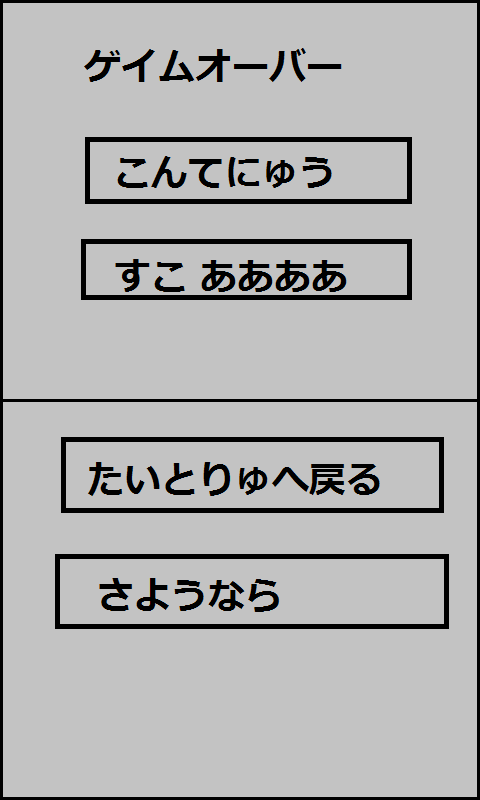
* 1. Javaファイル設計　option.java
     1. xml各オブジェクトの関連付け
     2. メニュー画面への遷移イベント
     3. マスクデータとして難易度変数int level保持
        1. 難易度変数 private int level は 0 易 1 普通 2 難
     4. EasyButtonを押すと int level に0を代入し、易の周囲に緑の〇を描画する
     5. NomalButton を押すと int level に1を代入し、普の周囲に緑の〇の描画をする
     6. HardButton を押すと int level に2を代入し、難の周囲に緑の〇の描画をする
     7. ScoreEraseButton を押すと、データベース内にあるスコアを全て消し、初期化を行う
     8. ReturnButton を押すと、menu.javaへ画面遷移を行う
  2. その他ファイル
     1. Option.png
     2. Onryou.png
     3. Level.png
     4. Easy.png
     5. Nomal.png
     6. Hard.png
     7. SocreErase.png
     8. Return.png

1. スコア表示画面
   1. Score.xml
      1. 縦長方向で表示する。画面は回転しない
      2. LinearLayout1のBackgroundcolorに金色っぽい色を指定
      3. 1stlabelに１：を入れる
      4. 1stNameTextFieldに１位の名前を入れる
      5. 1stScoreに１位のスコアを入れる
      6. LinearLayout2のBackgroundcolorに銀色っぽい色を指定
      7. 2ndlabelに２：を入れる
      8. 2ndNameTextFieldに２位の名前を入れる
      9. 2stScoreに2位のスコアを入れる
      10. LinearLayout3のBackgroundcolorに銅色っぽい色を指定
      11. 3rdlabelに3：を入れる
      12. 3rdNameTextFieldに3位の名前を入れる
      13. 3rdScoreに3位のスコアを入れる
      14. LinearLayout4のBackgroundcolorに赤色っぽい色を指定
      15. 4thlabelに4：を入れる
      16. 4thNameTextFieldに4位の名前を入れる
      17. 4thScoreに4位のスコアを入れる
      18. LinearLayout5のBackgroundcolorに青色っぽい色を指定
      19. 5thlabelに5：を入れる
      20. 5thNameTextFieldに5位の名前を入れる
      21. 5thScoreに5位のスコアを入れる
      22. LinearLayout6のBackgroundcolorに緑色っぽい色を指定
      23. 6thlabelに6：を入れる
      24. 6thNameTextFieldに6位の名前を入れる
      25. 6thScoreに6位のスコアを入れる
      26. LinearLayout7のBackgroundcolorに紺色っぽい色を指定
      27. 3thlabelに7：を入れる
      28. 7thNameTextFieldに7位の名前を入れる
      29. 7thScoreに7位のスコアを入れる
      30. LinearLayout8のBackgroundcolorに紫色っぽい色を指定
      31. 8thlabelに8：を入れる
      32. 8thNameTextFieldに8位の名前を入れる
      33. 8thScoreに8位のスコアを入れる
      34. LinearLayout9のBackgroundcolorにピンク色っぽい色を指定
      35. 9thlabelに9：を入れる
      36. 9thNameTextFieldに9位の名前を入れる
      37. 9thScoreに9位のスコアを入れる
      38. LinearLayout10のBackgroundcolorに黒色っぽい色を指定
      39. 10thlabelに10：を入れる
      40. 10thNameTextFieldに10位の名前を入れる
      41. 10thScoreに10位のスコアを入れる
      42. BackTitleButton最下部にタイトル画面へ戻るボタン



* 1. Score.java
     1. ゲイムオーバー画面から遷移される
     2. 遷移された直後に名前入力欄を出し、半角６文字以内で入力させる。
     3. SQLiteを使ってTable Score を作成し、id, name, scoreを作成、ランキングの名前とそのスコアを格納する。
     4. Table ScoreのScoreを降順に並べ、１０位以内のデータを取り出し、1stScoreなどのデータにそれぞれ名前やスコアを格納する
     5. BackTitleButtonを押すとタイトルへ遷移イベントを起こす
  2. その他ファイル

1. ゲイムオーバー画面
   1. GameOver.xml
      1. 画面中央で水平に分割
      2. 画面は縦に使用する
      3. gemeovertitle 上部にゲイムオーバーイメージ
      4. contenuebutton 上部にコンテニューボタン
      5. socrebutton 上部にスコアボタン
      6. titlebackbutton 下部にタイトルに戻るボタン
      7. exitbutton 下部にさようならボタン



* 1. GameOver.java
     1. Xml各オブジェクトの関連付け
     2. プレイ画面への遷移イベント
     3. スコア表示画面への遷移イベント　スコアを添える
     4. タイトル画面への遷移イベント
     5. アプリ終了イベント
     6. スコアを保持している
  2. その他ファイル
     1. gameover.png
     2. contenue.png
     3. score.png
     4. titleback.png
     5. exit.png

1. プレイ画面
   1. Play.xml
      1. 画面中央で水平方向に分割
      2. 画面は縦に使用する
      3. QLabel 問題画像用のラベル
      4. MissLabel 間違い画像用のラベル
      5. ScoreNameLabel 画面左上に配置するラベル
      6. ScoreLabel ScoreNameLabelの右に配置するラベル
      7. OptionButton 画面右上に配置するボタム
      8. TimeLabel 画面上に配置するラベル
      9. TimeIconLabel 画面上から画面端を時計回りに進めていくラノベ



* 1. Play.java
     1. xmlとの関連付け
     2. コンストラクター
        1. QLoadメソッドを呼び出す
     3. ランダム関数　rnd
     4. QSelect 問題決定メソッド
        1. ランダム関数rnd及び、画面遷移時に受け取った難易度変数levelを使用する
        2. 問題番号を引き出す
     5. QLoadメソッド
        1. 問題を読み込むメソッド
        2. 問題決定メソッド QSelect を呼び出す
        3. このメソッドが2回目以降も呼ばれた場合は 配列 int Used[ ]の中身にある番号との比較を行う
           1. 配列に番号が無かった場合はそのまま続行する
           2. 配列に番号があった場合は2の問題決定メソッド呼び出しに戻る
        4. 配列 int Used[ ] に問題番号を挿入する
        5. 問題番号に相応した画像2枚、間違い5か所の座標、5つの座標に対する誤差の許容範囲のアタイを取得する
           1. 座標は int point1 , int point2 , int point3 , int point4 , int point5に代入する
           2. 誤差はint accept [w][ww ]に代入する
           3. wには間違いポインツの番号1～5が挿入する
           4. wwには許容座標のxa,ya,xb,ybの順でそれぞれ1,2,3,4,で管理される
           5. ※例　1番目の間違い座標の許容範囲が左上(x1,y1)、右上(x2,y2)、左下(x3,y3)、右下(x4,y4)の間の場合

x1=x3

x2=x4

y1=y2

y3=y4

accept[1][1] = x1

accept[1][2] = x4

accept[1][3] = y2

accept[1][4] = y3

* + - 1. 取得した画像2枚を QLabel、MissLabelに代入する
    1. TimerEvent
       1. private int time で管理を行う
       2. final int maxtime = 200 で秒数200を最大値に設定
    2. Judge 判定メソッド
       1. TouchEvent で取得した座標が正解かどうかの比較を行う
       2. 指の大きさを考え誤差を考慮しなければならない
       3. 変数　private int kamo = 0 を用意する
       4. 変数　private int kamoxo = 0 を用意する
       5. 変数　private int Seiei = 0 を用意する
       6. accept[w][ww]との比較を行う
       7. ※accept[w][0]の値が1の場合は以下の処理を行わない
       8. TouchEventで取得した座標がwwから取得したx1,x2 y1,y2の間のアタイであれば正解とし、外れている場合は不正解とする
       9. accept[w][ww]のww = 0の部分に正解判定を挿入、1で正解となる
       10. 正解の場合、描画イベントを呼び出す。x1,x2 y1,y2, Boolean color = trueを送り付ける。
       11. 正解の場合、変数 kamo に 1を加算する
       12. 正解の場合、変数kamoxoとkamoを比較し、kamoxoが小さい場合、kamoxoにkamoを代入する
       13. 正解の場合、変数Seieiに1を加算する
       14. 不正解の場合、変数time – 20 の処理を行う。変数 kamoに0を代入する
    3. TouchEvent
       1. タッチ、タッチ、ここにタッチすると(x,y)で座標を取得する
       2. Judge 判定メソッドを呼び出す
    4. 〇描画イベント
       1. Canvasを使用
       2. 判定メソッドから受け取ったx1,x2 y1,y2, Boolean colorを使用する
       3. Boolean color = true の場合、緑色の描画となる
       4. Falseの場合は赤澤である
       5. 有色塗りつぶし描画をした後、若干範囲が小さくなった無色塗りつぶし描画を行う。残るのは線になるはず
       6. 緑色の描画を終了した時点でスコア加算メソッドを呼びつけ、10点の加点を行う
    5. End　問題児終了メソッド
       1. このメソッドは常に監視している
       2. Timerが0、またはSeieiが5となった時点で処理を開始する
       3. Seieiが5の場合、画面中央に次の問題に切り替えるボタンを表示させる
       4. This way…　このボタンを押すと新しい問題を読み込む
       5. Timerが0の場合、accept[w][ww]を引っ張ってきて、accept[1 ~ 5][0]を確認していく。[1 ~ 5][0]の中身が1以外の場合は不正解なので、booleadn color = false 、x1,x2,y1,y2を描画メソッドに送る
       6. Timerが0の場合、スコアを添えてゲイムオーバー画面へ移動する
       7. 更に、問題がすべてなくなった場合は、画面(クダン)中央に「おめでとうございます」と表示させる
       8. ↑の状態はタッチすると終了する。終了次第、そのままスコアを添えてゲイムオーバー画面へ移動する
    6. ElonaPlas スコア加算メソッド
       1. 判定メソッドのkamoxoとtimerを使用する
       2. Kamoxoは最大連続正解数である
       3. 50 \* 1.2 \* (kamoxo -1)
       4. Timer \* 2
       5. スコア合計-50 + (50 \* 1.2(kamoxo -1)) + (Timer \* 2)
  1. 冬のその他ファイル

1. クレジット画面
   1. Credit.xml
      1. 画面は縦に使用する
      2. gamenameTextField ゲーム名表示テキストフィールド
      3. versionTextField アプリバージョン表示テキストフィールド
      4. ReleaseTextField リリース年月日表示テキストフィールド
      5. DevelopmentteamTextField 開発チーム表示テキストフィールド
      6. titlebackButton 画面最下部にタイトル画面に戻るボタン
   2. Credit.java
      1. タイトル画面に戻るイベント
   3. その他ファイル