Table of Contents

Class Header

```
Zweck: In der Instanz dieser Klasse werden alle vom Spieler
nicht veränderbaren (Konstanten) Parameter gespeichert.
Die Modifikation der Parameter erfolgt durch den Programmierer
Weiter dient die Instanz dieser Klasse zum speichern aller
Prommablaufsteurungs releveanter Variablen

%   Class Name: GameParameter.m
%   Call: name = GameParameter()%
```

Changelog

- Version 00.00.01 07.10.15 Raphael Waltenspül Erstellt des Main Objekts, noch nicht Objekt Orientiert.
- Version 00.00.11 28.10.15 Joel Koch Code Aufgeräumt
- Version 00.01.00 22.11.15 Raphael Waltenspül Umbau in Objektoriert erfogt
- Version 00.01.02 10.12.15 Raphael Waltenspül Neu Erstellen der Handle Classes GameParameter, Gamestates, Wether
- Version 00.01.10 01.01.16 Raphael Waltenspül Neu Entwickeln der Buttons / Game Mode Taktik in Figure
- Version 00.01.11 02.01.16 Raphael Waltenspül Aufräumen fertigstellen Gameablauf
- Version 01.00.00b 03.01.16 Raphael Waltenspül Buglist Testen Kommentieren Dokumentieren

Input und Output

für Methoden, siehe Methoden

```
%   Konstruktor:   void
%   Precondition:
%
%   Postcondition: Ein GameStates Instanz ist erstellt
%
% Variables:
%       Für Instanzvariabeln siehe Properties
```

Implementierte Methoden

```
% this = GameStates()
% this = setMenueProccessed(this, state)
% state = getProcessState(this)
% [] = setPlayerInGame(this, number)
% [playerInGame] = decreasePlayerInGame(this)
% [playerInGame] = getPlayerInGame(this)
% [] = setActualPlayer(this,number)
% [actPlayer] = getActualPlayer(this)
% [] = nextPlayer(this, GameParameter, PlayerArray)
% [gameRound] = getGameRound(this)
% [gameRound] = nextGameRound(this, GameParameter)
```

Buglist TODO / this

Classdef

```
classdef GameStates < handle
    properties (GetAccess = public)

            FONT = 'Courier';    % Sans Schriftart für Ganzes Spiel
            FONT_SERIF = 'Times New Roman'; % Serif Schriftart für
 Ganzes Spiel
            TITLE_SIZE = 19;     % Textgrösse für Titel
            TEXT_SIZE = 15;      % Textgrösse Standard
            TEXT_SIZE_SMALL = 15; % Textgrösse Klein
            TEXT_SIZE_TINY = 13;  % % Textgrösse sehr Klein

            TITLE_COLOR = [.0,.1,.8]; % Titelfarbe
            GREEN = [.01, .5, .01];   % Stadnard Grün
            HOVER_GREEN = [.01, .7, .01] % Stadnard Grün für
 Hovereffekt Momentan nicht verwendet
            BLACK = [.01, .01, .01]; % Stadnard Schwarz
            BACK_BLACK = [.01, .01, .01]; %% Schwarz für Hintergrund
            RED = [0.8,0.1,0.15]; % Stadnard Rot
            ORANGE = [0.9,0.4,0.1]; % Stadnard Orange
            YELLOW = [0.9,0.9,0.1]; % Stadnard Gelb
            MAGENTA = [1,0,1]; % Stadnard Magenta
```

```
            SKY = [0.6 0.9 1]; % Hellblau Himmel

            varScreenSize = get(0,'ScreenSize'); % Bildschirmgrössen
            SCREEN_WIDTH; % Bildschirmbreite
            SCREEN_HIGH; % Bildschirmhöhe

            MENUE_HIGH;  % Menue höhe
            MENUE_WIDTH;  % Menue breite
            MENUE_POSITION;  % Menue position

            GAME_HIGH;  % Spiel höhe
            GAME_WIDTH;  % Spiel breite
            GAME_POSITION; % Spiel position
    end

    properties (Access = private)
```

Programmstatus

Mit folgenden Parameter wird der Status des Programess Beschrieben Anmerkung. Statemacheen wurde noch nicht weiter verfolgtt. Für Komplexere Versionen des Programmes vorgesehen

```
      menueProcessed = 0;

      actualPlayer = 1; % Spieler Welcher Am zug ist
      playerInGame; % Anzahl Spieler welcher noch im Spiel verbliben
sins

      gameRound = 1; % Die Aktuelle Spielrunde

    end

    methods

        function this = GameStates()
```

GameState Konstruktor

Zweck: Instanz von GameStates ist erzeugt

```
            % Pre:
            %
            % Post: GameStates ist erstellt
            %
            % Input:void
            %
            % Output: Instanz GameState
            %
            % Modifizierte Instanzvariable
            % this.SCREEN_WIDTH
            % this.SCREEN_HIGH
            % this.MENUE_HIGH
            % this.MENUE_WIDTH
```

```
            % this.MENUE_POSITION
            % this.GAME_HIGH
            % this.GAME_WIDTH
            % this.GAME_POSITION
            %
            this.SCREEN_WIDTH = this.varScreenSize(3);
            this.SCREEN_HIGH = this.varScreenSize(4);

            this.MENUE_HIGH = this.SCREEN_HIGH/8*5;
            this.MENUE_WIDTH = this.MENUE_HIGH/8*5;

            this.MENUE_POSITION = [this.SCREEN_WIDTH/2-
this.MENUE_WIDTH/2, ...
                this.SCREEN_HIGH/2 -
 this.MENUE_HIGH/2,this.MENUE_WIDTH, this.MENUE_HIGH];

            this.GAME_HIGH= this.varScreenSize(4);
            this.GAME_WIDTH= this.varScreenSize(3);

            this.GAME_POSITION = [ 0, 0, this.GAME_WIDTH,
 this.GAME_HIGH];

        end
```

GameState setMenueProccessed

Zweck: Setter zum setzen des Menue States

```
        % Pre: Instanz GameState ist erstellt
        %
        % Post: Menustate ist gesetzt
        %
        % Input: state -- sztatus des menues
        %
        % Output: Instanz GameState
        %
        % Modifizierte Instanzvariable
        %   this.menueProcessed
        %
        function this = setMenueProccessed(this, state)
            this.menueProcessed = state;
        end
```

GameState getProcessState

Wird noch nicht verwendet Zweck: Getter für die Statemachine

```
        % Pre: Instanz GameState ist erstellt
        %
        % Post: Status ist zurückgegeben
        %
        % Input: Instanz GameState, instanzvariabeln
        %
```

```
% Output: state -- Programmstatus ist zurückgegebn
%
function state = getProcessState(this)
    state = this.menueProcessed;
end
```

GameState setPlayerInGame

Zweck: Setter für die Anzahl Spieler im Spiel

```
% Pre: Instanz GameState ist erstellt
%
% Post: Anzahl Spieler ist gesetzt
%
% Input: number -- Anzahl Spieler
%
% Output: void
%
% Modifizierte Instanzvariable
%   this.playerInGame
%
function [] = setPlayerInGame(this, number)
    this.playerInGame = number ;
end
```

GameState decreasePlayerInGame

Zweck: Reduziert die Aktuelle anzahl Spieler im Spiel und gibt die neue Anzahl an Spieler zurück.

```
% Pre: Instanz GameState ist erstellt
%
% Post: Anzahl Spieler ist reduziert, die neue ANzahl ist
% zurückgegeben
%
% Input: Instanz GameState, instanzvariabeln
%
% Output: playerInGame -- neue Anzahl an Spilern
%
% Modifizierte Instanzvariable
%   this.playerInGame
%
function [playerInGame] = decreasePlayerInGame(this)
    this.playerInGame = this.playerInGame - 1;
    playerInGame = this.playerInGame;
end
```

GameState getPlayerInGame

Zweck: Getter für Anzahl Spieler im Spiel

```
% Pre: Instanz GameState ist erstellt
%
```

```
        % Post: Anzahl Spieler ist zurückgegeben
        %
        % Input: Instanz GameState, instanzvariabeln
        %
        % Output: playerInGame -- Anzahl Spieler welche noch im Spiel
 sind
        %
        function [playerInGame] = getPlayerInGame(this)
            playerInGame = this.playerInGame;
        end
```

GameState setActualPlayer

Zweck: Setter für Aktuellen Spieler

```
        % Pre: Instanz GameState ist erstellt
        %
        % Post: Aktueller Spieler ist gesetzt
        %
        % Input: number -- Aktueller Spieler
        %
        % Output: void
        %
        % Modifizierte Instanzvariable
        %   this.actualPlayer
        %
        function [] = setActualPlayer(this,number)
            this.actualPlayer = number;
        end
```

GameState getActualPlayer

Zweck: Getter für Aktuellen Spieler

```
        % Pre: Instanz GameState ist erstellt
        %
        % Post: Aktueller Spieler ist zurückgegeben
        %
        % Input: Instanz GameState, instanzvariabeln
        %
        % Output: actPlayer -- Aktueller Spieler
        %
        function [actPlayer] = getActualPlayer(this)
           actPlayer = this.actualPlayer;
        end
```

GameState nextPlayer

Zweck: Stellt den nächsten Spieler welchernoch über lebenspunkte verfügt ein

```
        % Pre: Instanz GameState ist erstellt
        % Instanz GameParameter ist erstellt
```

```
        % Instanz Player in PlayerArray sind erstellt
        %
        % Post: der nächste Spieler ist eingestellt
        %
        % Input: Instanz GameState, instanzvariabeln
        %  GameParameter --
        %  Player --
        %
        % Output: void
        %
        % Modifizierte Instanzvariable
        %   this.actualPlayer --
        %
        function [] = nextPlayer(this, GameParameter, PlayerArray)

            if GameParameter.playerQuantety == this.actualPlayer
                this.actualPlayer = 1;
            else
                this.actualPlayer = this.actualPlayer + 1;
            end
            watchdog = 0;
            while PlayerArray(this.actualPlayer).livePoints <= 0 ||
watchdog < 2 * GameParameter.playerQuantety
                if GameParameter.playerQuantety == this.actualPlayer
                    this.actualPlayer = 1;
                else
                    this.actualPlayer = this.actualPlayer + 1;
                end
                watchdog = watchdog + 1;
            end
        end
```

GameState getGameRound

Zweck: Getter für Spielrunde

```
        % Pre: Instanz GameState ist erstellt
        %
        % Post: Spielrunde ist zurückgegeben
        %
        % Input: Instanz GameState, instanzvariabeln
        %
        % Output: gameRound -- Aktuelle Spielrunde
        %
        function [gameRound] = getGameRound(this)
            gameRound = this.gameRound;
        end
```

GameState nextGameRound

Zweck: Stellt die nächste Spielrunde ein

```
        % Pre: Instanz GameParameter und GameState ist erstellt
```

```
        %
        % Post: neue Spielrunde ist eingestellt
        % Aktuelle Spielrunde ist zrückgegeben
        %
        % Input: Instanz GameState, instanzvariabeln
        %        GameParameter
        %
        % Output: gameRound -- Aktuelle Spielrunde
        %
        % Modifizierte Instanzvariable
        %   this.gameRound --
        %
        function [gameRound] = nextGameRound(this, GameParameter)
            this.gameRound = this.gameRound + 1;
            if  this.gameRound > GameParameter.numberRounds
                this.gameRound = 'End';
            end
            gameRound = this.gameRound;
        end

    end
end
```

Published with MATLAB® R2015b