Table of Contents

Class Header

Zweck: In der Instanz dieser Klasse werden Alle daten und Methoden zu den einzelnen Spielern bereitgestellt. Dies Beinhaltet Lebenspunket, Namen, Punktestand, sowie die Farbe und Grafischen objekte. Dies um die Spieler für kommende Anpassungen Varaiabel zu halten. (beispielsweise könnte jeder Spieler ein anderes Fahrzeug erhalten)

```
%   Class Name: Player.m
%   Call: name = Player(number, name, tankType, GameParameter)
```

Changelog

- Version 00.00.03 15.10.15 Joel Koch Panzer erstellt
- Version 00.00.10 25.10.15 Joel Koch Panzer Detailliert entworfn
- 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 00.01.03 11.12.15 Raphael Waltenspül Neu Erstellen der Handle Classes Figure, Player
- Version 00.01.05 20.12.15 Raphael Waltenspül Implementieren der Funktione Panzer in Player, neu Panzer Farbe, Intersection 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:     number -- Nummer des Spieler
```

```
%                     name -- Name des Spielers -- Verwendung in
%                     späteren Versionen
%                     tankType -- Name des Panzertyps -- Verwendung in
%                     späteren Versionen
%                     GameParameter -- Instanz von GameParameter
%
%   Precondition:  Instanz GameParameter ist erstellt
%                  Anzahl Spieler ist bekannt, Nummer übergeben
%
%   Postcondition: Eine Player Instanz ist erstellt
%
```

Implementierte Methoden

```
% this = Player(number, name, tankType, GameParameter)
% [] = genTank(this)
% [color] = getTankColor(this)
% [] = posTank(this, Landscape, GameParameter)
% [tankArray]= getTank(this)
% [angle] = calcAngle(this, mouseX, mouseY)
%
```

Buglist TODO / this

Classdef

```
classdef Player < handle
    properties(GetAccess = public)
        number; % Nummer des Spieler
        name; % Name des Spieler
        livePoints; % Lebenspunkte des Spieler
        score; % Spielpunkte des Spieler
        tankType; % Fahrzeugtyps des Spieler
        positionXY; % Position des Spieler
        tankArray;  % Zeichnung des Fahrzeugs
        tankHandler; % Handler der Zeichnung des Fahrzeugs
    end

    methods
```

Player Konstruktor

Zweck: Erstellt eine Instanz der Klasse Player

```
        % Pre: Instanz GameParameter ist erstellt
        %    Anzahl Spieler ist bekannt, Nummer übergeben
        %
        % Post: Instanz Player ist erstellt
        %
        % Input: Instanz Player, instanzvariabeln
        %   number -- Nummer des Spieler
```

```
        %   name -- Name des Spielers
        %   tankType -- Name des Panzertyps
        %   GameParameter -- Instanz von GameParameter
        %
        % Output: Instanz Player, instanzvariabeln
        %
        function this = Player(number, name, tankType, GameParameter)
            this.number = number;
            this.name = name;
            this.livePoints = GameParameter.getStandardLivePoints;
            this.score = 0;
            this.tankType = tankType;
            this.positionXY = [0,0];
        end
```

Player genTank

Zweck: Zeichnen eines Panzers als XY Array

```
        % Pre: Instanz Player ist erstellt
        %
        % Post: XY Array für Polygon ist erstellt
        %
        % Input: Instanz Player, instanzvariabeln
        %
        % Output: void
        %
        function [] = genTank(this)
            % Liefert [x y] des panzers.
            % wenn nr ungereade dann ist der panzer invers (Player L
und Player R)

            %Um den panzer zu vergrössern / verkleinern
             panzerScale = 0.5;
             % besserer Panzer Poligon
             x=[30 80 91 91 83 86 86 76 62 61 60 60 58 52 46 10 11 5
0  6  7  43 43 29 21 21 30];
             y=[0  0  6  14 14 17 19 28 28 30 30 34 34 30 26 48 49 53
47 43 44 22 15 14 11 8  0 ];
             x = panzerScale*x;
             y = panzerScale*y;
             x= x-max(x)/1.4;            % Panzerbody auf die
Playerpos zentrieren (horizontale)
             % Noch aus alter Version als Panzer immer am gleichen
             % ort standen

            if mod(this.number,2) ~= 0       %player spiegeln wenn der
Spieler rechts positioniert ist
                x=-x;
            end

            this.tankArray = [x;y]; % Tankrray speichern
        end
```

Player getTankColor

Zweck: Jedem Spieler eine eigene Farbe zuweisen

```
        % Pre: Instanz Player ist erstellt
        %
        % Post: [r g b] Array für Polygonfarbe ist Spielrspezifisch
 erstellt
        %
        % Input: Instanz Player, instanzvariabeln
        %
        % Output: color --  [r g b] Arra
        %
        function [color] = getTankColor(this)
                pk = this.number + 8; % binärcode 1000 erzeugen um
 auch führende Nullen zu erzwingen
                                      % die Spielernummer dazuaddien
                                      % Spieler 1 = 1001, Spieler 6 =
 1110
                % Für jeden Spieler ein eigenes Binärmuster der Farben
                % erzeugen
                r = dec2bin(pk);
                r = bin2dec(r(4));
                g = dec2bin(pk);
                g = bin2dec(g(3));
                b = dec2bin(pk);
                b = bin2dec(b(2));
                color =[r,g,b]; % Muster als RGB Werd zurückgeben
        end
```

Player posTank

Zweck: den Spieler zufällig in seinem Bereich plazieren. Das Polygon für die plazierung berechnen

```
        % Pre:  Instanz Player ist erstellt
        %       Instanz GameParameter ist erstellt
        %       Instanz Landscape ist erstellt
        %
        % Post: this.tankArray ist modifiziert und liegt auf der
 oberfläche
        % des Landscapes
        %
        % Input: Instanz Player, instanzvariabeln
        %       GameParameter --
        %       Landscape --
        %
        % Output: void
        %
        % modifizierte Instanzvariable
        %   this.tankArray --
        %
        function [] = posTank(this, Landscape, GameParameter)
            terrain = Landscape.getLandscape();
```

```
            if mod(this.number, 2) == 0
                value = length(terrain) - round(rand() *
length(terrain)* GameParameter.maxTankPos);
            else
                value = round(rand() * length(terrain)*
GameParameter.maxTankPos);
            end
            this.positionXY = terrain(:,value);

            this.tankArray(1,:) =  this.tankArray(1,:) +
this.positionXY(1,1);
            this.tankArray(2,:) =  this.tankArray(2,:) +
this.positionXY(2,1);
        end
```

Player getTank

Zweck: getteer Methode für das Tank Array

```
        % Pre:  Instanz Player ist erstellt
        %       this.tankArray ist erstellt
        %
        % Post: this.tankArray ist zurückgegeben
        %
        % Input: Instanz Player, instanzvariabeln
        %
        % Output: tankArray;
        %
        function  [tankArray]= getTank(this)
            tankArray = this.tankArray;
        end
```

Player getTank

Zweck: berechnen des Winkels zwischen maus und spieler

```
        % Pre:  mausposition ist bekannt
        %
        % Post: der relative Winkel zwischen Maus und Fahrzeug wurde
        % zurückgegeben
        %
        % Input: Instanz Player, instanzvariabeln
        %           mouseX -- Mausposition in X koordinaten
        %           mouseY -- Mausposition in Y koordinaten
        % Output: angle -- relative Winkel zwischen Maus und Fahrzeug
        %
        function [angle] = calcAngle(this, mouseX, mouseY)
            angle = atan((mouseY - this.positionXY(2,1))/(mouseX -
this.positionXY(1,1)));
            angle = abs((angle*180)/(pi()));
        end

    end
```

```
end
```

Published with MATLAB® R2015b