

---

## Table of Contents

Class Header .....	1
Changelog .....	1
Input und Output .....	1
Implementierte Methoden .....	2
Buglist TODO / this .....	2
Classdef .....	2
Wether Konstruktor .....	2
v von wind wird in m/s berechnetMittelwert 0 Normalverteilt .....	2
Wether updateWether .....	2
Wether getWindShape .....	3
Wether getWindShapeColor .....	3

## Class Header

Zweck: In Instanzen dieser klasse wird das Wetter für das Spiel erstellt und die änderung des Wetters während des Spiels berechnet. Weiter wird das Windpfeilshape in AAbhängikeit der Windstärke erstellt und dessen Farbe ermittelt.

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

## Changelog

- Version 00.00.13 12.11.15 Raphael Waltenspül Menu zur eingabe von Parametern erstellt
- Version 00.01.00 22.11.15 Raphael Waltenspül Umbau in Objektorientiert erfolgt
- Version 00.01.01 12.11.15 Raphael Waltenspül Menu zur eingabe von Parametern erweiter auf Objektorientiert
- Version 00.01.02 10.12.15 Raphael Waltenspül Neu Erstellen der Handle Classes GameParameter, Gamestates, Wether
- Version 00.01.11 02.01.16 Raphael Waltenspül Aufräumen fertigstellen Gameablauf
- Version 01.00.00b 03.01.16 Raphael Waltenspül Buglist Testen

## Input und Output

für Methoden, siehe Methoden

```
% Konstruktor: GameParameter
% Precondition: GameParameter sind Instanziert
%
% Postcondition: Instanz von FlightPath ist erstellt
%
% Variables:
```

---

```
%      Für Instanzvariablen siehe Properties
%
```

## Implementierte Methoden

## Buglist TODO / this

## Classdef

```
classdef Wether < handle
    properties
        gameParameter; % Instanz der GameParameter
        wind; % die Windstärke auf in x richtung
    end

    methods
```

## Wether Konstruktor

Zweck: Instanz von Wether ist erzeugt

```
% Pre: die GameParameter sind Instanziert und übergeben
%
% Post: Wether ist erstellt
%
% Input: GameParameter
%
% Output: Instanz Wether
%
% Modifizierte Instanzvariable
%   this.wind --
function [this] = Wether(GameParameter)
```

## v von wind wird in m/s berechnetMittelwert 0 Normalverteilt

```
        this.gameParameter = GameParameter;
        % die Windstärke wird Normalverteilt gerechnet unter
einbezug der einstelllungen
        this.wind = randn() * 10 * this.gameParameter.numberWind;

    end
```

## Wether updateWether

Zweck: Die Windstärke wird neu berechnet, dies in abhängikeit von der Aktuellen Windstärke

```
% Pre: Wether ist erstellt
%
```

---

```

    % Post: this.wind ist neu gerechnet
    %
    % Input: Wether Instanz
    %
    % Output: void
    %
    % Modifizierte Instanzvariable
    %   this.wind --
    function [] = updateWether(this)
        % die Änderung Windstärke wird Normalverteilt gerechnet
        unter einbezug der einstelllungen
        this.wind = this.wind + (this.wind / 100 * randn() * 10 *
        this.gameParameter.numberWind);
    end

```

## Wether getWindShape

Zweck: Das Windshape wird in Abhängigkeit der Windstärke berechnet, und Als [x,y] Array zurückgegeben.

```

    % Pre: Wether ist erstellt
    %
    % Post: windVektor ist zurückgegeben
    %
    % Input: Wether Instanz
    %
    % Output: windVektor --
    %
    function [windVektor] = getWindShape(this)
        % Das Windpfeilshape
        x = [0, 60, 60, 210, 210, 60, 60 ,0];
        y = [0 , -45, -15, -15 ,15, 15, 45, 0];
        x = x * this.wind / 50; % Das Windpfeilshape in
        abhängigkeit der Windstärke
        % Das Windpfeilshape wird in der Rechten oberen Ecke
        plazierte.
        x = x + this.gameParameter.PLOT_W - (max(x)+10);
        y = y + this.gameParameter.PLOT_H - (max(y)+10);
        windVektor = [x;y];
    end

```

## Wether getWindShapeColor

Zweck: Die Farbe des Windshape wird in Abhängigkeit der Windstärke berechnet, und Als [r,g,b] Array zurückgegeben.

```

    % Pre: Wether ist erstellt
    %
    % Post: windVektor ist zurückgegeben
    %
    % Input: Wether Instanz
    %
    % Output: windShapeColor --

```

---

```
%
function [windShapeColor] = getWindShapeColor(this)
    % Hier wird die Farbe des Windpfeilshapes ermittelt
    % Diese geht von Rot in Grün über
    if this.wind > 60;
        windTemp = 1;
    else
        windTemp = abs(this.wind/60);
    end
    r = windTemp;
    g = 1-windTemp;
    b = 0;
    windShapeColor = [r,g,b];
end

end

end
```

*Published with MATLAB® R2015b*