**Game engine documentation**

**This documentation was generated automatically from the xml style comments of the project.**

**Library used: SFML library (Link: https://www.sfml-dev.org/download/sfml/2.5.1/)**

**<?xml version="1.0"?>**

**<doc>**

**<assembly>**

**"GameEngine"**

**</assembly>**

**<members>**

**<member name="M:Snake.HandleSnakeOutsideMap">**

**<summary>**

**This method handles the situation where the snake gets outside of the playing area**

**</summary>**

**</member>**

**<member name="M:Snake.ResetSnake">**

**<summary>**

**Reset the snake**

**</summary>**

**</member>**

**<member name="M:Snake.CheckDeath">**

**<summary>**

**Check if the snake died**

**</summary>**

**<returns>True if the snake touched its tail false otherwise</returns>**

**</member>**

**<member name="M:Snake.Draw(sf.RenderWindow\*)">**

**<summary>**

**Draw the snake**

**</summary>**

**<param name="window">The window is being drawn to</param>**

**</member>**

**<member name="M:Snake.Eat">**

**<summary>**

**This function gets called when the snake eats food**

**Adds one more to the tails length and changes the foods position**

**</summary>**

**</member>**

**<member name="M:Snake.SetDirection(System.Int32,System.Int32)">**

**<summary>**

**Set the snakes x, y movement direction with the given parameters**

**</summary>**

**<param name="x">Given x direction</param>**

**<param name="y">Given y direction</param>**

**</member>**

**<member name="M:Snake.Move">**

**<summary>**

**Move the snake in the x,y direction**

**</summary>**

**</member>**

**<member name="F:Snake.snakeSpeed">**

**<summary>**

**The speed at which the snake is moving**

**</summary>**

**</member>**

**<member name="F:Snake.xDirection">**

**<summary>**

**The x,y movement direction of the snake**

**</summary>**

**</member>**

**<member name="F:Snake.tail">**

**<summary>**

**A list of sprites representing the snakes tail**

**</summary>**

**</member>**

**<member name="F:Snake.food">**

**<summary>**

**A Snake food object**

**</summary>**

**</member>**

**<member name="F:Snake.texture">**

**<summary>**

**The texture used to create the head and the tail sprites**

**</summary>**

**</member>**

**<member name="F:Snake.sprite">**

**<summary>**

**The sprite that represents the head of the snake**

**</summary>**

**</member>**

**<member name="M:Snake.Update(sf.RenderWindow\*)">**

**<summary>**

**The update loop called once a frame**

**</summary>**

**<param name="window">A pointer to the render window</param>**

**</member>**

**<member name="M:Snake.Start">**

**<summary>**

**The first function that gets called after the snake is created**

**</summary>**

**</member>**

**<member name="F:SnakeFood.texture">**

**<summary>**

**The texture used to create the sprite**

**</summary>**

**</member>**

**<member name="F:SnakeFood.sprite">**

**<summary>**

**The sprite to be drawn**

**</summary>**

**</member>**

**<member name="M:SnakeFood.SetLocation">**

**<summary>**

**Sets the foods location to a random location on the screen**

**</summary>**

**</member>**

**<member name="T:SnakeFood">**

**<summary>**

**Struct that represents the food eaten by the snake**

**</summary>**

**</member>**

**<member name="M:Random.RandomNumberInRange(System.Int32,System.Int32)">**

**<summary>**

**Generate a random integer in the given range**

**</summary>**

**<param name="min">The maximum number allowed (included)</param>**

**<param name="max">The minimum number allowed (included</param>**

**<returns />**

**</member>**

**<member name="T:Random">**

**<summary>**

**Class that it's used to generate random numbers**

**</summary>**

**</member>**

**<member name="F:Input.keysByState">**

**<summary>**

**Holds all the buttons in the SFML Key enumeration**

**True indicates the key has been released, false the key is still pressed**

**</summary>**

**</member>**

**<member name="M:Input.UpdateKeyMap(sf.Keyboard.Key,System.Boolean)">**

**<summary>**

**Updates the key map with the given state**

**</summary>**

**<param name="button">The button that has its state changed</param>**

**<param name="state">Bool that represents the state of the given button</param>**

**</member>**

**<member name="M:Input.GetButtonUp(sf.Keyboard.Key,sf.RenderWindow\*)">**

**<summary>**

**Checks if the button was released this frame or not.**

**</summary>**

**<param name="button">Pressed button</param>**

**<param name="window" />**

**<returns>True the frame the button was released, false otherwise</returns>**

**</member>**

**<member name="M:Input.GetMouseButtonUp(sf.Mouse.Button,sf.RenderWindow\*)">**

**<summary>**

**Checks if the mouse button was released this frame or not.**

**</summary>**

**<param name="button">Pressed mouse button</param>**

**<param name="window" />**

**<returns>True the frame the button was released, false otherwise</returns>**

**</member>**

**<member name="M:Input.GetButtonDown(sf.Keyboard.Key)">**

**<summary>**

**Checks if a button was pressed in the frame it was called.**

**</summary>**

**<param name="button">Pressed button</param>**

**<param name="window" />**

**<returns>True the frame the button was pressed, false otherwise</returns>**

**</member>**

**<member name="M:Input.GetMouseButtonState(sf.Mouse.Button)">**

**<summary>**

**Checks a mouse button's state.**

**</summary>**

**<param name="button">The enumeration value of the checked button</param>**

**<returns>Return true every frame the mouse button is pressed and false every frame the mouse button is not pressed</returns>**

**</member>**

**<member name="M:Input.GetButtonState(sf.Keyboard.Key)">**

**<summary>**

**Checks a button's state.**

**</summary>**

**<param name="button">The enumeration value of the checked button</param>**

**<returns>Returns true every frame the button is pressed and false every frame the button is not pressed</returns>**

**</member>**

**<member name="T:Input">**

**\defgroup graphics Graphics module**

**2D graphics module: sprites, text, shapes, ...**

**<summary>**

**Input management class**

**</summary><summary>**

**Input management class**

**</summary></member>**

**</members>**

**</doc>**