Funktionsbeschreibungen

## Vorlage:

|  |  |  |
| --- | --- | --- |
| Name | | Modul |
| Beschreibung |  | |
| Input |  | |
| Output |  | |

## Grafikfunktionen:

|  |  |  |
| --- | --- | --- |
| DrawTransformedImage | | Grafik.c |
| Beschreibung | Draws the given image scaled and rotated at the given position into the current image. | |
| Input | ImageId: Handle of image to draw x, y position to draw image at…  Scalex: Scalingfactor for x axis (float value)  Scaley: Scalingfactor for y axis (float value)  Angle: Angle to rotate Image (in rad) | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_sharp\_empty\_rectangle | | Grafik.c |
| Beschreibung | Draws empty rectangle with sharp edges. | |
| Input | x and y koord. as windowskoord.,  int Width, int Height, ColorType Color, int LineWidth | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| pixel\_to\_map | | Grafik.c |
| Beschreibung | Convert windowskoord. to map position. | |
| Input | x and y as windowskoord. | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| map\_to\_pixel | | Grafik.c |
| Beschreibung | Convert mappositon to windowskoord. (Point upper left) | |
| Input | x and y as map position | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_playground | | Grafik.c |
| Beschreibung | Draws a playground (field and lines) | |
| Input |  | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| scale\_handler | | Grafik.c |
| Beschreibung | Returns the percentage for scaling the image to field size. | |
| Input | Image\_ID, a valid ID of a loaded image file | |
| Output | size scale,  x- and y-scale factor in percentage of the field size | |

|  |  |  |
| --- | --- | --- |
| draw\_focus | | Grafik.c |
| Beschreibung | Draws a focus at the selected field. | |
| Input | x and y as map position | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_rot\_focus | | Grafik.c |
| Beschreibung | Draws a rotation-image on the selected field | |
| Input | x and y as map position | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_empty\_field | | Modul |
| Beschreibung | Deletes the selected field (reset). | |
| Input | x and y as map position | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_half\_laser | | Grafik.c |
| Beschreibung | Help-function for draw\_laser and draw\_angled\_laser.  Draws half of the laser in the selected field  (v1.1: Laser glows). | |
| Input | x and y as map position, direction | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_laser | | Grafik.c |
| Beschreibung | Draws the laser in the selected field (2x draw\_half\_laser). | |
| Input | x and y as map position, direction | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_angled\_laser | | Grafik.c |
| Beschreibung | Draws the angled laser in the selected field (2x draw\_half\_laser). | |
| Input | x and y as map position, direction, angle (-1 right, 1 left) | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| destroy\_images | | Grafik.c |
| Beschreibung | Deletes with init\_images() loaded images from memory | |
| Input |  | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| init\_images | | Grafik.c |
| Beschreibung | Loads images from files into memory | |
| Input | Blup… | |
| Output | Blup… | |

|  |  |  |
| --- | --- | --- |
| draw\_figure | | Grafik.c |
| Beschreibung | Draws figure at its location with its rotation/direction | |
| Input | pawn \*figure (figurepointer) | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_figure\_destroyed | | Grafik.c |
| Beschreibung | Draws/animates the destruction of a mirror.  (V1.0, it only draws an empty field  (V1.1, "Melting"-animation with rectangles  (V1.2, offset increases always 1 pixel, not laser width  (V1.3, New animation, with glow) | |
| Input | pawn \*figure | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_invert\_colors | | Grafik.c |
| Beschreibung | Inverts the colours of the defined part. | |
| Input | x and y as windowskoord. for start position; width and height for the size | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| draw\_winner\_text | | Grafik.c |
| Beschreibung | Writes winner text on screen. | |
| Input | pawn \*hit\_king | |
| Output |  | |

## Logikfunktionen:

|  |  |  |
| --- | --- | --- |
| laser | | Logik.c |
| Beschreibung | Draws the laser from the cannon across the whole playground and calls all the other functions handling figure behaviour. | |
| Input | Receives the field from which the laser shoot is done. this  field is not painted with laser anymore, but the field NEXT to it, whose direction from the field is specified with dir. | |
| Output | If a wall or a cannon was hit, or the laser passes out of the playground, return 0. If a king was hit: -1 for player\_red, -2 for player\_blue.  If a mirror was hit: +1 for player\_red, +2 for player\_blue.  (In case of a splitter being hit: then two laser paths generated and the return value is the one with the priority (descending order): king, mirror, wall / cannon) | |

|  |  |  |
| --- | --- | --- |
| is\_inside\_map | | Logik.c |
| Beschreibung | Checks if the given coordinates are inside the array. | |
| Input | Given coordinates (map position) | |
| Output | If inside map (means, inside the range [0 - 7][0 - 5], then it returns 1.  Otherwise it returns 0 | |

|  |  |  |
| --- | --- | --- |
| is\_figure | | Logik.c |
| Beschreibung | Checks if the given coordinates (map position) contains a figure. | |
| Input | x and y as map position | |
| Output | If there is a figure, return 1.  If it’s an empty field, return 0  (A wall is threatened as a figure). | |

|  |  |  |
| --- | --- | --- |
| move\_figure | | Logik.c |
| Beschreibung | Moves a figure to the given location. | |
| Input |  | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| destroy\_figure | | Logik.c |
| Beschreibung | Destroys a figure (deletes it from the map array). | |
| Input | pawn \*figure | |
| Output |  | |

|  |  |  |
| --- | --- | --- |
| mouseclick\_to\_map | | Logik.c |
| Beschreibung | Get Mouse-Clicks and returns the map position. | |
| Input |  | |
| Output | Returns location struct, of the field who was hit or ERROR when the click was beyond the map or there was no click. | |

|  |  |  |
| --- | --- | --- |
| path\_handler | | Logik.c |
| Beschreibung | Combines the two strings path and file after checking if there's enough memory available. | |
| Input | const char path[] - String with the path of file  char file[] - String with the filename | |
| Output | returns string with the complete path | |

|  |  |  |
| --- | --- | --- |
| play\_sound | | Logik.c |
| Beschreibung | Plays the sound of chosen enumeration. | |
| Input | Enumeration: Laser, Reflexion, Destruction, Victory, Ignore, Intro, Music, Bling, Bell. | |
| Output |  | |

## GRAFIK

(*pos* ist ein struct beschrieben in LaserChess.h, welches x und y koordinate enthält)

### pixel\_to\_map(pos)

Erhält pixel-koorinaten, gibt map-position zurück (es wird einfach auf das feld darunter ‚gerundet‘)  
Gibt x und y = -1 wenn pixel ausserhalb der map.

### map\_to\_pixel(pos)

Erhält map-koorinaten, gibt obere linke koordinate in windowskoorinaten zurück

### draw\_laser(pos, dir)

Zeichnet eine linie in map-position x, y. und zwar von der richtung dir aus zum gegenüberliegenden rand des feldes.

### draw\_angled\_laser(pos, dir, angle)

Zeichnet einen 90° Winkel in map-position x, y. und zwar von der richtung dir aus, mit winkel angle auf eine der beiden nebenseiten. -> angle = -1 entspricht CW / rechts. Angle = +1 entspricht CCW / links.

### draw\_figure(\*figure)

Zeichnet an der Position x y die Figur \*figure

### draw\_mirror\_destroyed(\*figure)

Zeichnet eine Zerstörung eines Spiegels (wie auch immer, anfangs nur schwarzes feld).

### draw\_king\_destroyed(\*figure)

Zeichnet eine Zerstörung eines Kings (wie auch immer, anfangs nur schwarzes feld).

### draw\_playground()

Zeichnet einen schwarzen Bildschirm und die Gitterlinien darauf

### draw\_empty\_field(pos)

Zeichnet an x, y ein schwarzes Feld

### draw\_focus(pos)

### init\_map(int nr, struct \*figure[], struct \*figure[][])

## LOGIK / HELPER TOOLS

### is\_inside\_map(pos)

Gibt 0 zurück wenn ausserhalb der map, 1 wenn innerhalb der map.

### is\_figure(pos)

Gibt 0 zurück, wenn dort an x, y ein NULL-Pointer ist. Sonst 1.

### NORM(int dir) *(Makro)*

Normiert die Richtung dir in 0 – 3 (wenn grösser als 4: %=4, wenn kleiner als 0, +=4)

### move\_figure(\*figure, pos)

Zeichnet am alten ort ein schwarzes feld, am neuen ort die Figur ‚figure‘. Ändert ALS EINZIGE FUNKTION die Position im Struct der Figures!

## HAUPTMODULE

### main()

Mainfunktion

### spiel(\*map)

Erhält die Map als Argument, steuert das gesamte Spiel in einer Schlaufe

### Laser(pos, dir)

Zeichnet den Laser nach dem Spielzug und behandelt alle verschiedenen Szenarien. Ist rekursiv.

Rückgabewerte:

* 1 und 2: Spiegel von Spieler 1 oder 2 zerstört
* 0: Wand getroffen
* -1 und -2: König von Spieler 1 oder 2 zerstört

# Arbeitsaufteilung

Cyril: Laserfunktion

Jascha: Helpertools/Doku

Marcel: main, spiel

Nicola: Grafik