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 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 are generated and the return value is the one with the higher 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** | figure pointer, new playground location | |
| **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. Switches all sounds off, if a sound was not found. | |
| **Input** | Enum: Laser, Reflexion, Destruction, Victory, Ignore, Intro, Music, Bling, Bell. | |
| **Output** |  | |

**Spielfunktionen:**

|  |  |  |
| --- | --- | --- |
| **create\_focus** | | **Spiel.c** |
| **Beschreibung** | Draws a green Background on all free Fields around the  selected figure. | |
| **Input** | location struct (x-y-coordinates of selected figure) | |
| **Output** |  | |

|  |  |  |
| --- | --- | --- |
| **clear\_focus** | | **Spiel.c** |
| **Beschreibung** | Draws an empty field to all marked fields. | |
| **Input** | location struct (x-y-coordinates of selected figure) | |
| **Output** |  | |

|  |  |  |
| --- | --- | --- |
| **spiel** | | **Spiel.c** |
| **Beschreibung** | Handles the game: Treats the mouse inputs, execute laser(), displays winner, close graphics. | |
| **Input** | pawn \*figure (used for cannon-position) | |
| **Output** |  | |

**Mainfunktionen (Laserchess):**

|  |  |  |
| --- | --- | --- |
| **create\_figures** | | **LaserChess.c** |
| **Beschreibung** | Initializes all figures from a received figure array.  Sets figures to the default map-position.  Currently initializes 14 figures (Optional: Splitter not defined in this version). | |
| **Input** | Pointer to the original figure array in the main-procedure | |
| **Output** |  | |

|  |  |  |
| --- | --- | --- |
| **menu** | | **LaserChess.c** |
| **Beschreibung** | User can choose between the modes: ‘NORMAL’, ‘SETMODE’ and ‘quit the game’. | |
| **Input** |  | |
| **Output** | Enum Mode | |

|  |  |  |
| --- | --- | --- |
| **Set\_figure\_positions** | | **LaserChess.c** |
| **Beschreibung** | The player can set his pawn freely on the map.  The figures in the array are sorted by color.  To toggle the player: i/2 for red and (i/2)+7 for blue. | |
| **Input** | Pointer to Array of all the figures. | |
| **Output** | Returns -1 if exit button is pressed, otherwise 0. | |

|  |  |  |
| --- | --- | --- |
| **init\_game** | | **LaserChess.c** |
| **Beschreibung** | Initializes the game.  Calls the graphic-functions and places the figures. | |
| **Input** | Array of all the figures, play mode (to decide whether to place all the figures, to initialized state or let the users place them alternating). | |
| **Output** | If wild failure appears, returns 0, otherwise returns 1. | |

|  |  |  |
| --- | --- | --- |
| **clear\_map\_array** | | **LaserChess.c** |
| **Beschreibung** | Clears the map array (writes all positions to NULL). | |
| **Input** |  | |
| **Output** |  | |

|  |  |  |
| --- | --- | --- |
| **argument\_handler** | | **LaserChess.c** |
| **Beschreibung** | Reads the start-arguments. If EXE was started by opening a map file, it tries to load and start a game.  If there are start-variables defined, they will be set.  Unknown arguments are printed to screen. | |
| **Input** | int argn, number of arguments; char\* args[], arguments; pawn \*figure,  Figure-array needed to start a game. | |
| **Output** |  | |

|  |  |  |
| --- | --- | --- |
| **gfxmain** | | **LaserChess.c** |
| **Beschreibung** | Uber-main function. Will be called FIRST! | |
| **Input** | System console call parameters. (OS specific) | |
| **Output** |  | |