**ALLEGRO GAME PROJECT REPORT**

**(GO GAME)**

**Ahmad Alkasemi: 2018556304**

The project that I designed was solely done by me without the help of anyone. I.E the group that I joined didn't help me with anything because they didn't care about this subject.

The game I designed is called **GO GAME**.

The main goal of the game is making your "Region of leverage" bigger than that of your opponent's.

At first, you might get confused about where to put your stones because you could almost place it anywhere on the board.

One of the ways to spreading your leverage over a region is to "take over" the opponents stones.

The surrounded stones get removed from the board and handed over to opposite side as hostages.

Every hostage equals one point (when the points of leverage are added up in the end of the game.)

In the beggining you'll be shown the main screen and it consists of 'start' and 'how to play' and 'quit' options.

Moreover, when you start the game the confrontation begins and every stone from a certain type that eats another stones earns a point.

There are 3 rules: from **4 sides**, **3 sides** and **2 sides**.

**Examples of the rules:**

- **4 sides**:The black stone sits in the middle and when surrounded by white stones it gets removed and it counts as a point for the white stones team.

- **3 sides**: The black stone sits on the edge of the board and when surrounded by 3 white stones it gets removed and it counts as a point for the white stones team.

- **2 sides**: the black stone sits in one of the corners of the board and when surrounded by 2 white stones it gets removed and it counts as a point for the white stones team.

**When the board is filled with stones the winner gets announced and the points shown for each team.**

There are 3 potential ending screens **at the end:**

- when the white team's points are higher than those of the black team, "White Player Wins".

- when the black team's points are higher, "Black team wins".

- when both teams are equal on points a "Draw Match" screen pops up.

**Notes:**

\* All of the designs were done by me using the following softwares:

-Adone photoshop

-Adobe illustrator

\* For the first and second rule when surrounding the last stone we double click to delete the stone

**ADDONS WE USED IN OUR PROJECT**

1. **Image Addon**

This addon is used for implementing the images to main menu, game background and selection menus.

1. **Primitives Addon**

We utilized this addon to draw the lines which are required to create the game boards

1. **Audio Addon**

We used this addon to load sounds and musics that played; with the “you win text” when the level is completed, while the game is playing and to warn when the player makes a mistake

1. **Audio Codec Addon**

We utilized this addon to register the known audio file type handlers

1. **Font Addon**

We used this addon to arrange the types and sizes of the fonts,to put 1’s and 0’s to the screen

1. **TrueType Font Addon**

To load a font type from a file

1. **Color Addon**

To be able to arrange the colour in game with the help of ”RGB” function

**LIBRARIES WE USED**

* **stdio.h**

This is the standart initial input/output library of C language.

* **allegro5/allegro.h**

This library enables us to use Allegro functions.

* **allegro5/allegro\_font.h**

We used this library to be able to reach the Allegro fonts and their attributes. It enabled us to arrange the font types, sizes and forms.

* **allegro5/allegro\_ttf.h**

This library, we used to implement the TrueType Font from a file and its properties in our game. We used this to implement the images into our codes to be able to print the 1’s and 0’s.

* **allegro5/allegro\_image.h**

This library is required to use the images in Bitmap formats to richen our game visually. We used those Bitmap forms on the backgrounds of the menus and tell the player that he/she won when the game is completed without any violation or mistake.

* **allegro5/allegro\_primitives.h**

This library is needed to activate the primitives’s usage and set the geometric shapes and their attributes. We utilized it to draw the lines and set their thickness, color, starting point and ending points, also to draw the squares to vanish the previous element when pressed twice in a square on the game boards

* **allegro5/allegro\_audio.h**

We implemented this library in order to be able play the audio files. We played audios while game is playing, when a violation or a mistake made and when game is played correctly.

* **allegro5/allegro\_acodec.h**

This library enables us to register the known audio file handlers.

**ALLEGRO FUNCTIONS WE USED**

**al\_install\_keyboard**

To activate keyboard keys to use during the game. We used this function to be able to quit the game when pressed “Esc”.

**al\_install\_mouse()**

Used for activating the mouse events.

**al\_install\_audio()**

Used for initializing the audio files to play in the game.

**al\_init\_ttf\_addon()**

We utlilized this function be able to take a specific font type from a file. Called after "al\_init\_font\_addon" to make "al\_load\_font" recognize “.ttf” and other formats supported by “al\_load\_ttf\_font”.

**al\_init\_font\_addon()**

Used for utilizing the font types which exist as default in Allegro.

**al\_init\_image\_addon()**

We used this function to implement the images into our game.

**al\_init\_primitives\_addon()**

We used this function to be able to add the geometric shapes. We utilized this when putting squares and drawing the lines.

**al\_init\_acodec\_addon()**

This is the function that we used the properties of the acodec library in order to register the known audio files. Registers all the known audio file type handlers for "al\_load\_sample, al\_save\_sample, al\_load\_audio\_stream", etc.

**al\_reserve\_samples(2)**

Reserves a number of sample instances, attaching them to the default mixer. If no default mixer is set when this function is called, then it will create one and attach it to the default voice. If no default voice has been set, it, too, will be created.

**al\_set\_new\_display\_flags(ALLEGRO\_FULLSCREEN\_WINDOW)**

We used to set screen flags and when it is used together with "ALLEGRO\_FULLSCREEN\_WINDOW" it makes the created window fill the screen.

**al\_load\_sample**

We utilized this function to be able to load the audio files in our game based on their extension.

**al\_create\_sample\_instance**

We utilized this funciton in order to create a sample instance, using the supplied sample data.

**al\_set\_sample\_instance\_playmode**

This function sets the playmode of the sample instance specified (loop in this case).

**al\_attach\_sample\_instance\_to\_mixer**

Attaches a sample instance to a mixer. The instance must not already be attached to anything.

**al\_load\_ttf\_font**

We benefitted this function to implement a font type from a file.

**al\_load\_bitmap**

We used this to load an image file into a new ALLEGRO\_BITMAP. The file type is determined by the extension.

**al\_get\_default\_mixer**

We used this function, so that it returns the default mixer.

**al\_flip\_display()**

To reflect the changes properly on the screen. Copies or updates the front and back buffers so that what has been drawn previously on the currently selected display becomes visible on screen.

**al\_wait\_for\_event**

To wait for user to do any event. Waits until the specified event queue is non-empty.

**al\_draw\_bitmap()**

To print the specified Bitmap on the screen.

**al\_clear\_to\_color**

To make the screen of the color which is selected after erasing the all on the screen.

**al\_draw\_text**

To print a text on the speicified region of the window.

**al\_map\_rgb**

This is the function that used to arrange the colors of items by utilizng red, green and blue colors’ amounts.

**al\_play\_sample(click, 1, 0, 1, ALLEGRO\_PLAYMODE\_ONCE, NULL)**

This function is made to play the specified audio file. Plays the sample on one of the sample instances created by "al\_reserve\_samples".

**al\_stop\_sample\_instance**

This function is made to stop an sample instance playing.

**al\_uninstall\_keyboard**

We used this function to inactivate the keyboard events which were activated before in order not to face any problem when the game played again after closing and reopening.

**al\_uninstall\_mouse**

This function uninstalls the active mouse driver, if any. This will automatically unregister the mouse event source with any event queues.

**al\_destroy\_display**

We used this function o remove the open window.

**al\_destroy\_event\_queue**

We utilized this function in order to end the event taking from the player. Destroys the event queue. All event sources currently registered with the queue will be automatically unregistered before the queue is destroyed.

**al\_destroy\_font**

Used this function to uninstall the fonts that used while playing.

**al\_destroy\_bitmap**

We utilized this to remove the Bitmap formatted images.

**al\_destroy\_sample**

We used this to function to remove/uninstall the audio files.

**al\_set\_target\_bitmap**

This function selects the bitmap to which all subsequent drawing operations in the calling thread will draw to.

**al\_convert\_mask\_to\_alpha(board, al\_map\_rgb())**

We utilized this function in order to convert the given mask color to an alpha channel in the bitmap.

**al\_set\_target\_backbuffer**

This function selects the bitmap to which all subsequent drawing operations in the calling thread will draw to.

**al\_draw\_line**

We benefitted this function to draw a line segment between two points.

**al\_draw\_filled\_rectangle**

We utilized this function to a filled rectangle in specified X-axis and Y-axis.

------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------