**Statement of language used**

Processing-java which is an object-oriented language and is used to show graphical images and animations

**Naming of all necessary libraries**

Processing

**System diagram**

**Class GraphicalObject**

**-**x

-y

-my\_width

-my\_height

+isMouseOver()

+render()

+checkMousePressed()

+getType()

**Class Button extends GraphicalObject**

-my\_text

-my\_textSize

-paddingX

-paddingY

-textColour

-baseColour

-hoverColour

-outline

+checkMousePressed(menu, hasSomethingBeenPressed)

+reset()

**Class Menu extends GraphicalObject**

-my\_buttons

-my\_structures

-my\_textBox

-my\_menus

-my\_image

-hasImage

-hasTextBox

-hasStructures

-hasButtons

-hasMenus

-my\_text

-startText

-textX

-textY

-hasBackground

-backgroundColour

-outlineColour

-exitMenu

+reset()

+isMouseOverElement()

+getInput()

+setString(newString)

+isTextBoxFocused()

+getTextBox()

**Class TextBox extends GraphicalObject**

-isFocused

-inputText

-showCursor

-cursorDelay

-cursorPosition

-visibleText

-inputTextStartPos

-changeTextStartPos(changeBy)

-updateVisibleText()

+update()

+clear()

+inputKey(inpKey)

+setFocused(temp)

+getInput()

+getIsFocused()

+reset()

**Class GraphicalStructure extends GraphicalObject**

-structureID

-my\_cellSize

-textX

-textY

**Class GraphicImage extends GraphicalObject**

-img

**Structure**

-structure

-rotatedStructure

-gridX

-gridY

-my\_width

-my\_height

-my\_RWidth

-my\_RHeight

-name

+update()

+place()

+placeInLocation(x,y)

+getWidth()

+getHeight()

+getX()

+getY()

+get(x, y)

+getName()

+resetRotated()

+rotate(r)

**Full outline of each function**

|  |  |  |
| --- | --- | --- |
| Function Name | Parameters | What it does |
| setup | ------- | Sets up the background and the board. Also sets up the menu and GUI. |
| draw | ------- | Render the game and updates it |
| god | List of entities | Checks if there is a pattern, that causes an entity to be created or destroyed and change their colour to signify that. Then returns the new list |
| setupMenus | ------- | This sets up the menus array |
| setupStructures | ------- | This sets up the structures array |
| setupStructureMenu | ------- | This returns the structure menu (creating all the GraphicalStructures) |
| setupSaveGameMenu | ------- | This returns the open save game menu |
| setupOpenGameMenu | ------- | This returns the open game menu |
| setupGUI | ------- | This returns the GUI menu |
| setupMainMenu | ------- | This returns the Main menu |
| resetToDefaults | ------- | This restores all the default settings |
| cancelPlacement | ------- | This is the code that is run to cancel the placement of the structure and go back to the GUI menu |
| changeMenu | menuIndex | This changes the current menu that is being displayed |
| checkMousePressed | -------- | This senses when a mouse is pressed and checks if it has pressed any interactive objects |
| mouseWheel | event | This senses if the mouse wheel changes and rotates the structure or zooms in and out. |
| keyPressed | -------- | Senses if a key is pressed and does the necessary thing depending on the conditions of the game |
| keyReleased | -------- | Senses when a key is released and resets the variables to false, that would have been changed by keyPressed |
| checkKeys | -------- | This checks if any keys have been pressed and, if any of the arrow keys have been pressed, move the screen location |
| readFromFile | filename | This returns a Boolean[][] that has been read from a file. |
| saveToFile | Filename, struct | This saves a Boolean[][] to a file |
| openSavedGame | Filename | This opens a file and sets it to the board. Also checks if file exits |
| saveGame | Filename | This saves the current board to a file |
| readFromFile | Filename | This reads from a file and turns it into a 2d array that then can be used to place in later on. |
| clearBoard | -------- | This makes all the values of the board set to false (representing an empty cell location) |
| randomBoard | -------- | This randomises the Board |
| setBoardToStruct | struct | This sets the board to a structure given (The structure is automatically centred) |
| startGame\_Explore | -------- | This runs the randomBoard and starts the game |
| startGame\_sandbox | -------- | This starts the game with nothing |
| startGame\_file | -------- | This starts the game from a saved file |
| render | -------- | This controls all the rendering |
| renderBoard | -------- | This renders the board |
| mouseOverButton | -------- | This sets the cursor to a hand |
| mouseOverText | -------- | This sets the cursor to the text icon |
| mousePlacing | -------- | This sets the cursor to a cross |
| mouseNormal | -------- | This sets the cursor to the default |
| GraphicalObject.isMouseOver | -------- | This returns whether the mouse is over the object |
| GraphicalObject.render | -------- | This is a function that is overloaded for each child class – but it renders the object |
| GraphicalObject.  checkMousePressed | -------- | This is a function that is overloaded for each child class – but it checks if the mouse is pressed on the object |
| GraphicalObject.getType | -------- | This returns the type |
| Button.reset | -------- | This resets the button to its defaults |
| Button.checkMousePressed | menu, a boolean | This is basically the parent class version however it takes in some parameters |
| Structure.update | -------- | Updates the gridX and gridY position |
| Structure.place | -------- | Places the structure on the board at its current location |
| Structure.getWidth | -------- | Returns the width of the structure |
| Structure.getHeight | -------- | Returns the height of the structure |
| Structure.getX | -------- | Returns the gridX |
| Structure.getY | -------- | Returns the gridY |
| Structure.get | Int x, int y | Returns a particular cell in the structure |
| Structure.rotate | r | This updates the rotatedStructure variable and rotates the structure 90 degrees in a curtain direction |
| Structure.resetRotated | -------- | This sets the rotation back to 0 |
| Structure.getName | -------- | This returns the name of the structure |
| Structure.placeInLocation | x,y | This places the structure in a given co-ordinate |
| Menu.reset | -------- | This resets everything in the menu to its defaults |
| Menu.isMouseOverElement | -------- | This returns true or false to say weather the mouse is over an element |
| Menu.getInput | -------- | This gets the input from the TextBox – if it has one |
| Menu.setString | newString | This sets the string in the menu to a given one |
| Menu.isTextBoxFocused | -------- | This returns if the TextBox is focused (and returns false if it doesn’t have a textbox. |
| Menu.getTextBox | -------- | This returns the textbox, however it may be a nullptr |
| TextBox.update | -------- | This updates if the cursor should be shown – for the flashing effect. |
| TextBox.clear | -------- | This clears the textbox inputString |
| TextBox.inputKey | inpKey | This takes in a character and adds it to the inputText |
| TextBox.setFocused | newFocused | This sets the isFocused variable |
| TextBox.getInput | -------- | This returns the inputText variable |
| TextBox.getIsFocused | -------- | This returns weather it is focused |
| TextBox.reset | -------- | This resets the TextBox to the defaults |
| TextBox.changeTextStartPos | changeBy | This changes the start position of the visible text, making sure it take up the whole textbox |
| TextBox.updateVisibleText | -------- | This updates the visibleText variable. |

**Full outline of the global variables**

|  |  |
| --- | --- |
| Variable name | Description |
| SCREEN\_HEIGHT | Stores the screen height |
| SCREEN\_WIDTH | Stores the screen width |
| screenSpeed | Controls how fast the screen moves |
| backgroundColour | Stores the background colour |
| currentMenu | Stores the current menu |
| BOARD\_HEIGHT | Stores the board height |
| BOARD\_WIDTH | Stores the board width |
| ORIGINAL\_SCREEN\_GRID\_HEIGHT | Stores the default number of cells that fit in the height |
| ORIGINAL\_SCREEN\_GRID\_WIDTH | Stores the default number of cells that fit in the width |
| START\_GRID\_X | Stores the starting x position of the grid |
| START\_GRID\_Y | Stores the starting y position of the grid |
| cellSize | Stores the cell width/height in pixels |
| screenXPos | Stores the screen X position |
| screenYPos | Stores the screen Y position |
| screenGridHeight | Stores the current number of cells that fit in the height |
| screenGridWidth | Stores the current number of cells that fit in the width |
| mode | Stores an integer representing the mode the game is in |
| paused | Stores a Boolean controlling if the game is paused |
| board | Stores the board as a Boolean[][] |
| boardcopy | Stores a copy of the board, for the god function |
| timeControl | This controls how often the game is updated |
| menus | This stores all the menus that can be used |
| structures | This stores all the structures |
| currentStructureActive | This stores the current structure that is being placed |
| renderStructure | This stores a Boolean to say if the structure should be rendered on the board |
| upPressed | This stores whether up arrow key is pressed |
| downPressed | This stores whether down arrow key is pressed |
| rightPressed | This stores whether right arrow key is pressed |
| leftPressed | This stores whether left arrow key is pressed |
| shiftPressed | This stores whether shift key is pressed |
| mousePressedDelay | This stores an integer to control the delay on the mouse clicks |

**Sketch of User Interface**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Structures |  |  |  |  |  |  |  |  | Menu |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Play |  |  |  |  |  |  |  |  |  |