|  |
| --- |
| ConnectFourGUI |
| + COLOUR: tuple of int  + SQUARESIZE: int  + NUM\_COLS: int  + NUM\_ROWS: int  + WIDTH: int  + HEIGHT: int  + SIZE: tuple of int  + RADIUS: int  + DISPLAY: Surface  + columns\_list: tuple of Buttons  + board: ConnectFourBoard |
| + \_init\_(self: ConnectFourGUI, board: ConnectFourBoard(int, int)): NoneType  + update\_screen(self: ConnectFourGUI, mouse\_position: tuple of int): NoneType  + game\_over(self: ConnectFourGUI, winner: str): NoneType  + play\_game(self: ConnectFourGUI): NoneType  + reset\_board(self: ConnectFourGUI): NoneType  + draw\_board(self: ConnectFourGUI): NoneType  + update\_column\_buttons(self: ConnectFourGUI, mouse\_position: tuple of int): NoneType  + decide\_column(self: ConnectFourGUI, mouse\_position: tuple of int): int |

|  |
| --- |
| Main |
| + board: ConnectFourBoard  1  + GUI: ConnectFourGUI |
| + \_init\_(self: Main, GUI: ConnectFourGUI)  + get\_board(self: Main): ConnectFourBoard  + get\_GUI(self: Main): ConnectFourGUI |

1

­­­­­­

1

|  |
| --- |
| Screen |
| - height: str  - width: str  - colour: tuple of int  - scene: str |
| + \_init\_(self: Screen, height: int, width: int, colour: int, scene: str): NoneType  + get\_height(self: Screen): int  + get\_width(self: Screen): int  + get\_colour(self: Screen): tuple of int  + get\_current\_scene(self: Screen): str  + set\_height(self: Screen, height: int): NoneType  + set\_width(self: Screen, width: int): NoneType  + set\_colour(self: Screen, colour: tuple of int): NoneType  + set\_scene(self: Screen, current\_scene: str, next\_scene: str)  + remove\_scene(self: Screen, scene: str)  + save\_scene(self: Screen, board: ConnectFourBoard) |

1

|  |
| --- |
| TextBox |
| - text: str  - x: int  - y: int  - re\_font: str  - size: int  - display: Surface  - bold: bool  - colour: tuple of int  - italicize: bool |
| + \_init\_(text: str, pos\_x: int, pos\_y: int, re\_font: str, size: int, screen: Surface, bold=False, colour = (0,0,0), italicize=False)  + create\_text(self: TextBox): NoneType  + place\_text(self: TextBox): NoneType |

|  |
| --- |
| Disk |
| + NUMOFDISKS: int  - width: int  - x: int  - y: int  - line\_thickness: int  - colour: tuple of int |
| +\_\_init\_\_(self: Disk, width: int, x: int, y: int, line\_thickness: int, colour: tuple of int): NoneType + get\_width(self: Disk): int  + set\_width(self: Disk, width: int): NoneType  + get\_x(self: Disk): int  + set\_x(self: Disk, x: int): NoneType  + get\_y(self: Disk): int  + set\_y(self: Disk, y: int): NoneType  + get\_line\_thickness(self: Disk): int  + set\_line\_thickness(self: Disk, line\_thickness: int): NoneType  + get\_colour(self: Disk): tuple of int  + set\_colour(self: Disk, colour: tuple of int): NoneType |

|  |
| --- |
| Button |
| - text: str - surface: Surface - colour: tuple of int - x: int - y: int - width: int - height: int - pattern: str - colourofText: str - fontSize: int - clicked: bool |
| +\_\_init\_\_(self: Button, surface: Surface, text: str, colour: tuple of int, indicatorColour: str, colourOfText: str, x: int, y: int, width: int, height: int, pattern: str, fontSize: int): NoneType + place(self: Button): NoneType |

|  |
| --- |
| ConnectFourBoard |
| + dim\_row: int  + dim\_col: int  + p1: str  + p2: str  + em: str  + turn: str  + board: list of int |
| + \_init\_(self: ConnectFourBoard, rows: int, cols: int): NoneType  + drop(self: ConnectFourBoard, col: int): NoneType  + can\_drop(self: ConnectFourBoard, col: int): Bool  + other\_player(self: ConnectFourBoard): str  + whos\_turn(self: ConnectFourBoard): str  + switch\_turn(self: ConnectFourBoard): NoneType  + check\_for\_win(self: ConnectFourBoard): Bool  + check\_win\_at\_position(self: ConnectFourBoard, row: int, col: int): Bool  + alternation(self: ConnectFourBoard, x: int, y: int, dx: int, dy: int): Bool  + valid\_move(self: ConnectFourBoard, col: int, row: int): Bool  + is\_game\_over(self: ConnectFourBoard): Bool  + get\_drop\_loc(self: ConnectFourBoard, column: int): int |

2…3

0…42

1

1…6

1

1

1

1