|  |  |  |  |
| --- | --- | --- | --- |
| Class Name: | Description | Variables | Methods |
| board | This class is the board that all the objects are in this board. This is the main class that this program run at; and this class set the background of this game.  This class contains all the main methods like checking the collision and key listeners. | Player h (instance) : the object that player can control to push the box  Box item (instance): the box that the player is pushing with. It is used to check the collision and the completed situation.  Int Upper(instance): the y intercept of the object, it is used to determine the y – intercept of object when initially created.  Int side(instance): the x intercept of the object, it is used to determine the x – intercept of object when initially created.  Int velX(instance): the distance of movement in x line direction of player  Int velY(instance): the distance of movement in y line direction of player  Arraylist walls : the list that contains all wall objects  ArrayList boxes(instance): the list that contains all box objects  Arraylist areas (instance): the list that contains all area objects which are the area that boxes should be placed  Frame map(instance): the frame that contains all object  JPanel contentPanel(instance): the panel contains all the objects | 1. Void keyPressed (KeyEvent event): this method is used to take action if the user presses up, down, left and right on the keyboards.   Parameter: keyevent event: the keyevent the player pressed  Return : nothing   1. Boolean collision : it is used to check whether the player meets the wall   Parameter: int a: the type of direction: 1 means left, 2 means right, 3 means up, 4 means down  Return : true/false   1. Int check : it is used to check whether the player meets a box and the box whether meets the wall   Parameter : the integer a : the type of direction: 1 means left, 2 means right, 3 means up, 4 means down  Return : int 1/2 : 1 means the box meets the wall  2 means box meets another box     1. Void paint : it is used to paint all the objects on the screen   Return: nothing  Parameter: graphics g |
| box | It is the class that contains all states and methods for the box including its position (x&y), and the image it has. | Int x (instance) : the x-coordinate of the box  Int y (instance): the y-coordinate of the box  Image a(instance) : the image of the box | **Public image getP():**  **It is used to get the image from the file**  **Return : nothing**  **Parameter: nothing**  **Public void setImage()**  **It is used to change the image of the box**  **Return : nothing**  **Parameter: nothing** |
| player | It is the class that defines the player including its position (x&y) and its image | Int x(instance) : the x-coordinate of the player  Int y(instance) : the y-coordinate of the player  Image a (instance): the image of the player | **Public image getP():**  **It is used to get the image from the file** |
| area | It is the class that defines the area including its position (x&y) and its image | Int x (instance): the x-coordinate of the area  Int y(instance) : the y-coordinate of the area  Image a(instance) : the image of the area | **Public image getP():**  **It is used to get the image from the file**  **Return: nothing**  **Parameter: nothing**  **Public void setImage()**  **It is used to change the image**  **Of the area.**  **Return: nothing**  **Parameter: nothing**  **Public empty image(){**  **It is used to change the image into null**  **Return: nothing**  **Parameter: nothing** |
| Wall | It is the class that defines the wall including its position (x&y) and its image | Int x(instance) : the x-coordinate of the wall  Int y (instance): the y-coordinate of the wall  Image a(instance) : the image of the area | **Public image getP():**  **It is used to get the image from the file** |
| level | It is the class that has all levels just like a collection of all maps | Arraylist levelList (static) : the list contains all the levels  String level1: the level1 map of the game represented by the string  String level2: the level2 map of the game represented by the string | **addLevel(): add the level to the list (static)**  **return: nothing**  **parameter: nothing**  **getLevel(): get the level from the list**  **return: nothing**  **parameter : nothing** |