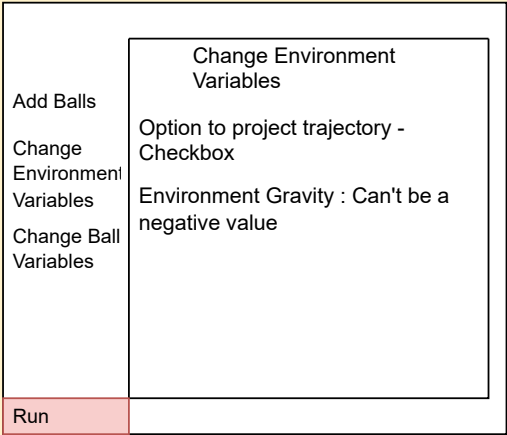
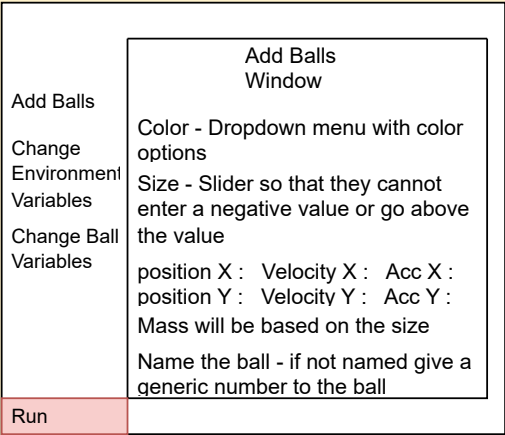
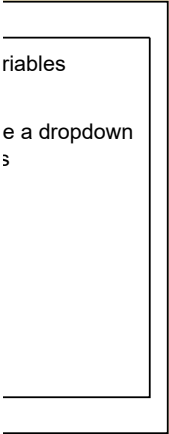
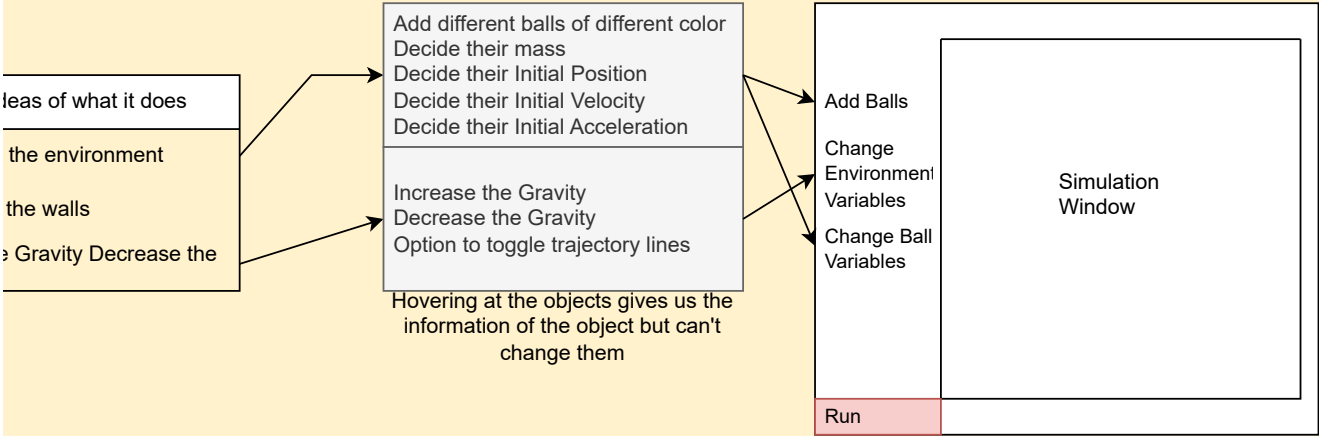
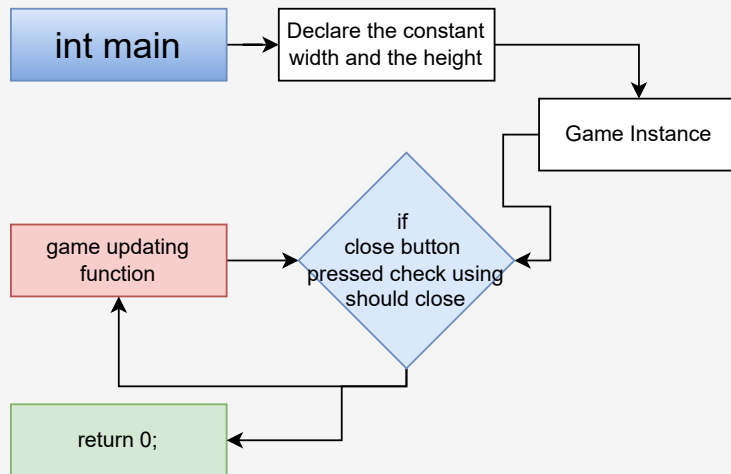


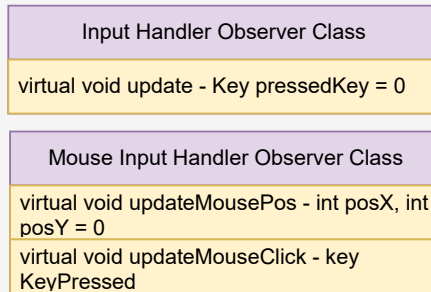
# Ball and Gravity Simulator



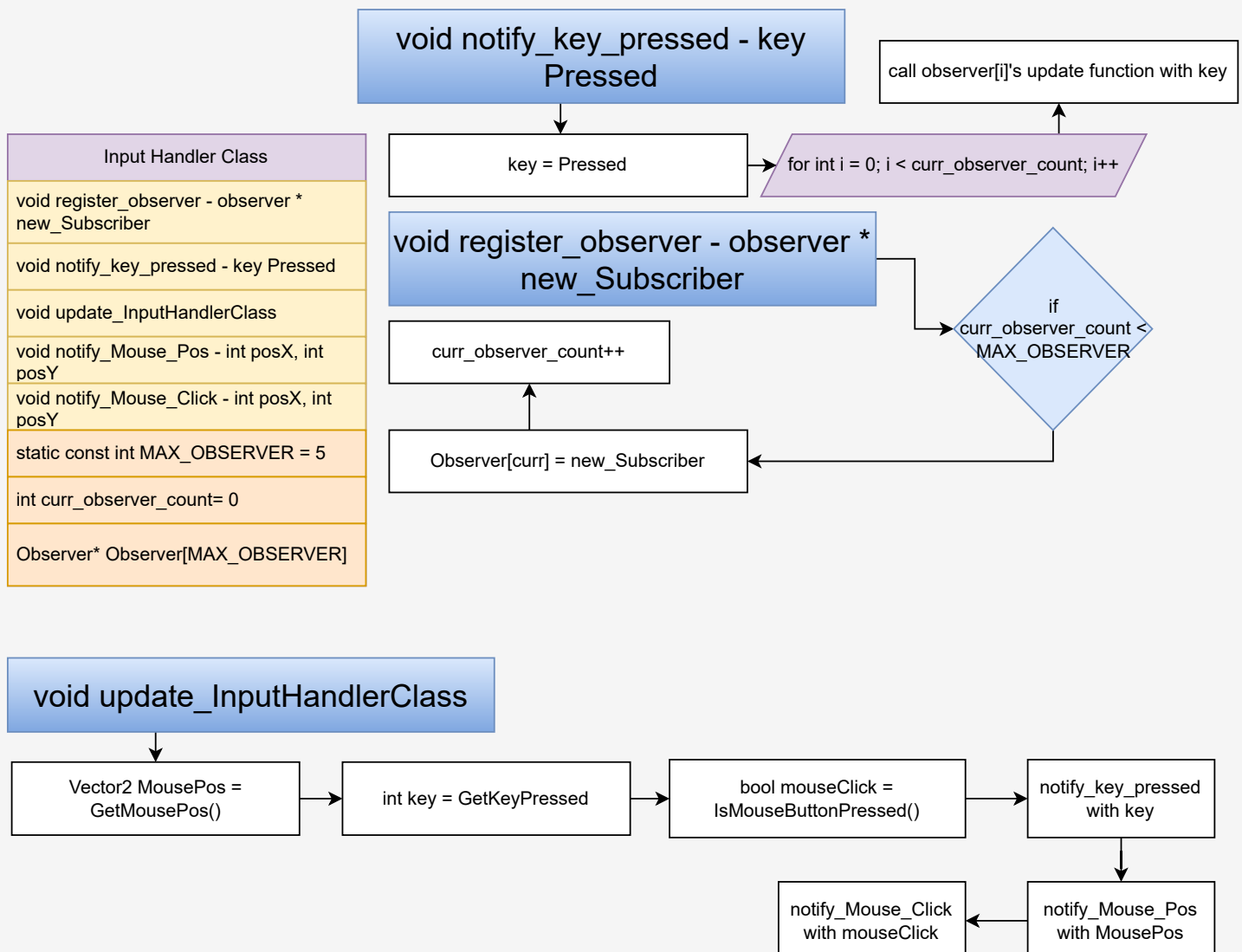
# Main Function



# Observer System for Input



# Input Handler System



# Game System

Game System
Constructor - int width, int height
void tick
bool should_close
void draw
void update
Input Handler Class Input
Destructor

Default Constructor = Delete

Constructor - int width, int height

Init the window with width,  
height and "Gravity  
Simulator"

Set Target FPS to 60

void tick

update

tick

bool should\_close

return shouldWindowsClose()

void draw

void update

Call Input's  
update\_InputHandlerClass

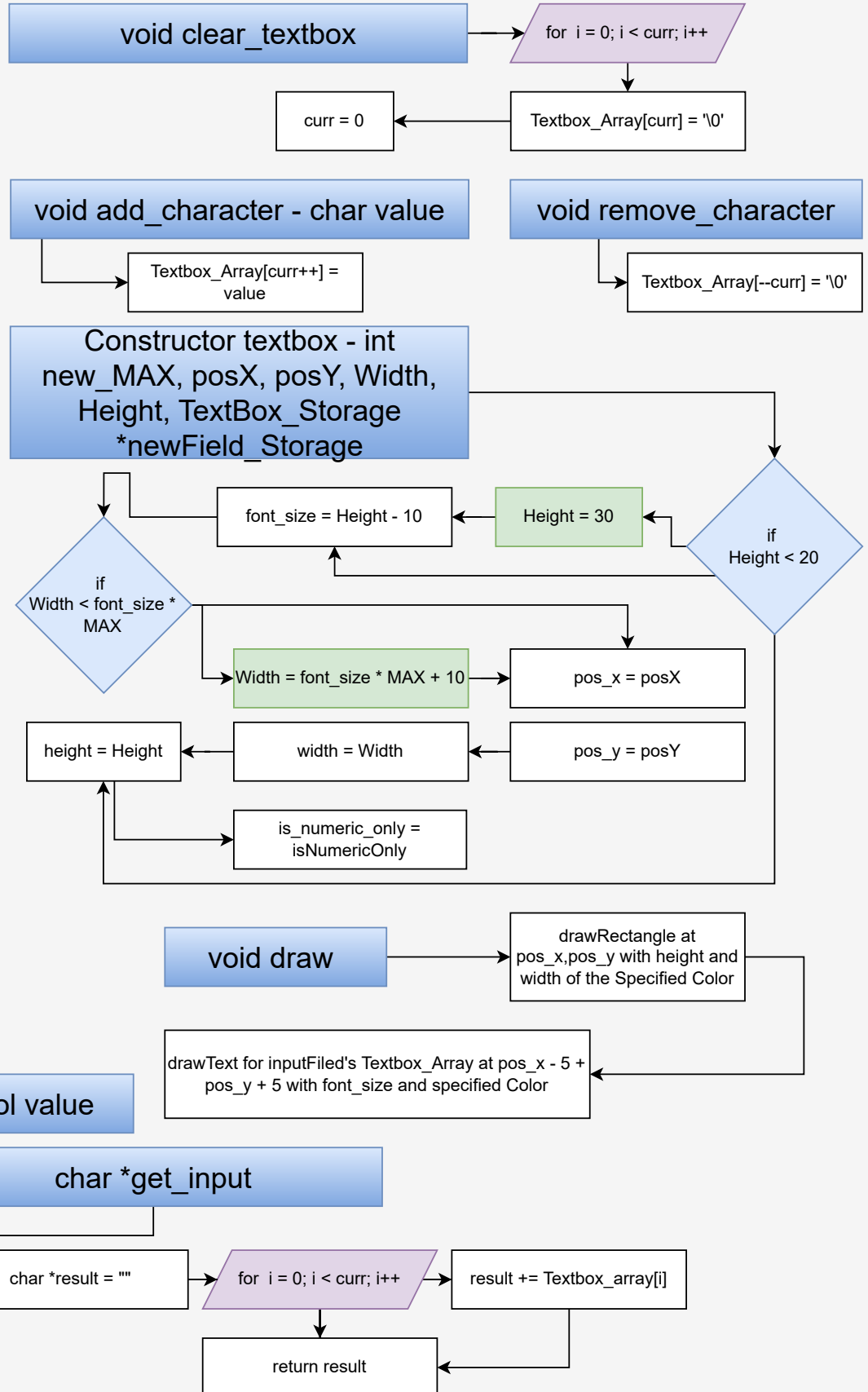
# TextBox System

```

Class TextBox_storage
static const int MAX = 100
int curr = 0
char* Textbox_Array[MAX]
void clear_textbox
char *get_input
void add_character - char value
void remove_character
    
```

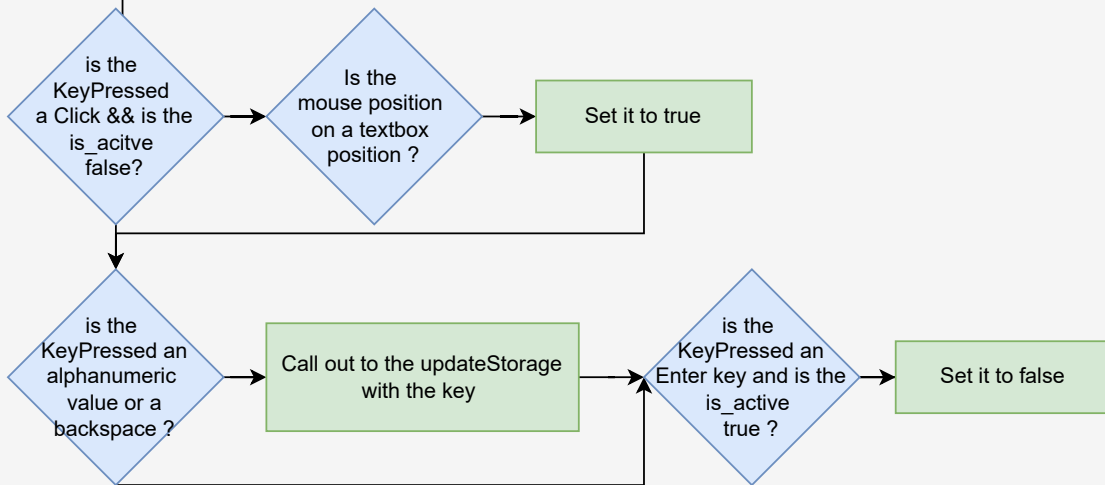
```

Class textbox: public Input Handler
Observer
Constructor textbox - posX, posY, Width, Height, bool isNumericOnly = false
void draw
void update - key KeyPressed override
void updateStorage - key KEY
void set_numeric - bool value
int char_to_int_convertor
Destructor textbox
TextBox_Storage *inputField
char *input
int pos_x, pos_y, width, height, font_size
bool is_active, is_numeric_only
    
```

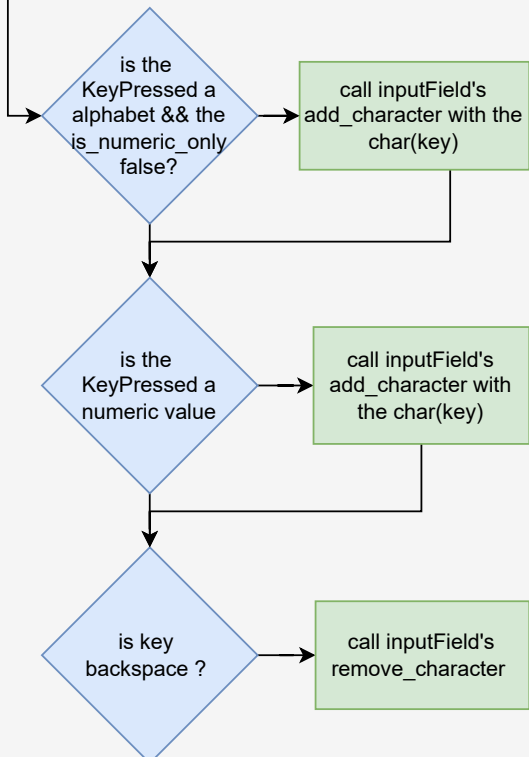


# TextBox System

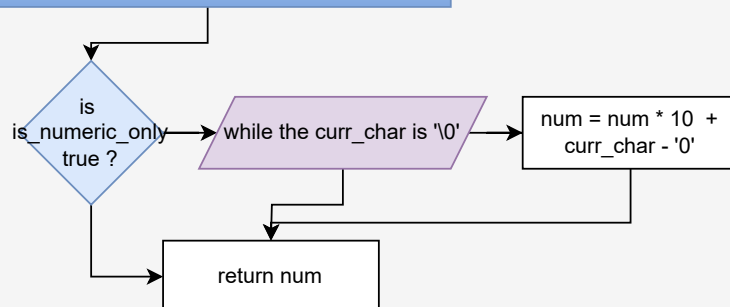
void update - key KeyPressed  
override



void updateStorage - key  
KEY

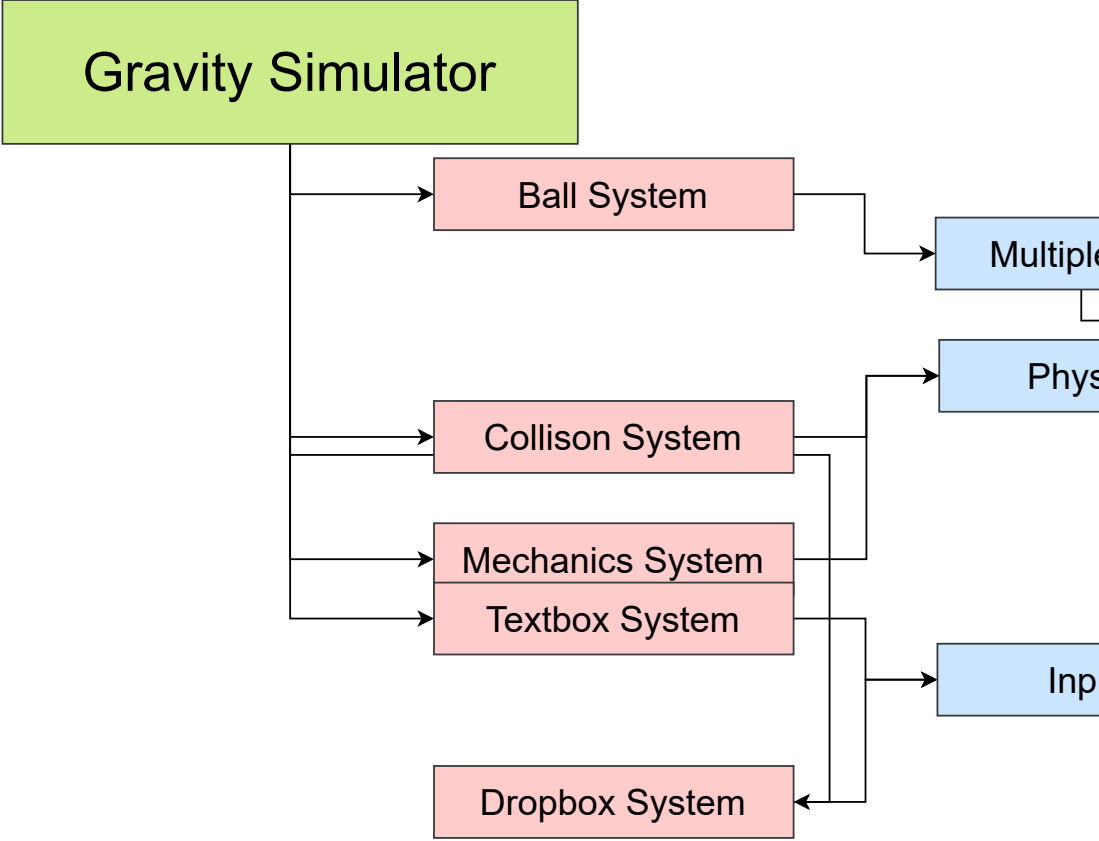


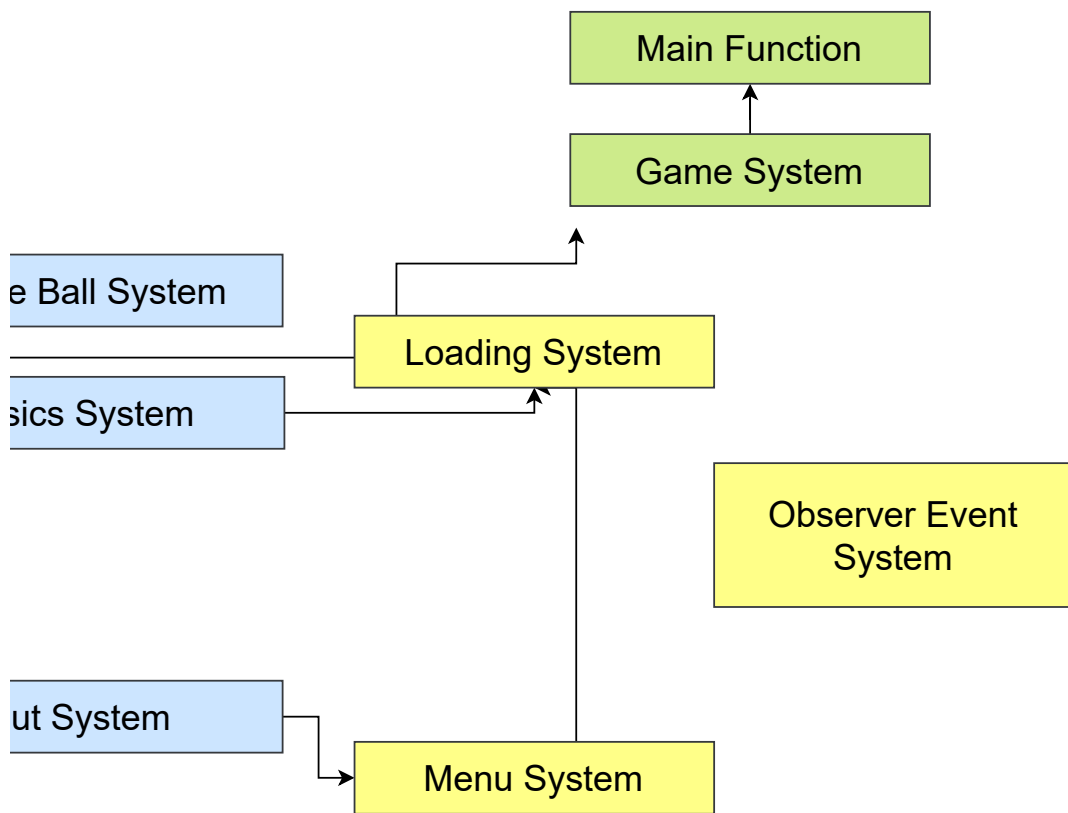
int char\_to\_int\_convertor



Destructor Textbox

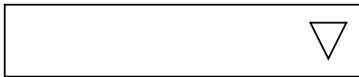
call inputField's clear  
textbox function





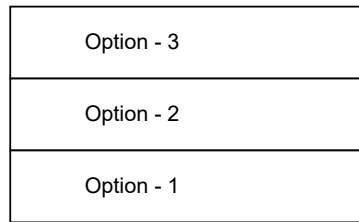
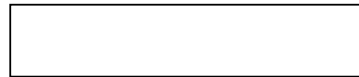


# DropDown System



## Dropdown UI

If the user clicks on the down arrow below, then all the options are displayed as seen in the following concept figure.

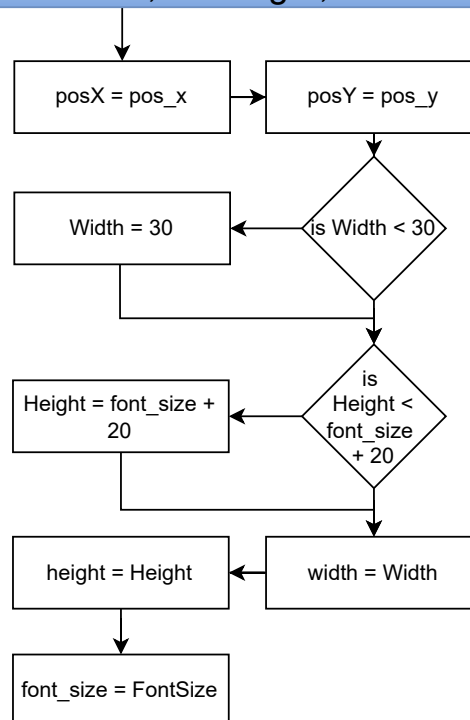


## Dropdown UI

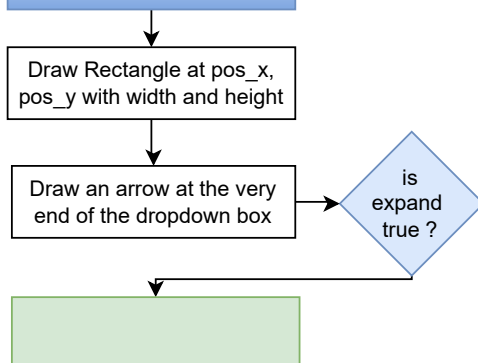
The thing expands and presents all the options

Class Dropdown - template M_TYPE: public Mouse Input Handler Observer Class
Constructor - posX, posY, Width, Height, FontSize
void update_Dropdown_menu
void draw _Dropdown_menu
void populate - M_TYPE new_option
void remove - M_TYPE target
M_TYPE select_option
virtual void updateMousePos - int posX, int posY = 0
virtual void updateMouseClicked - key KeyPressed
Destructor
int pos_x, pos_y, width, height, font_size
bool expand

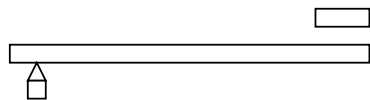
## Constructor - int posX, int posY, int Width, int Height, int FontSize



## void draw



# Slider System



## Slider UI

The slider system will let the user slide and then they it will return a value and then it will be stored in a variable and then we will use setter functions. As well as the textbox at the bottom will store the value there is not specific need to integrate it with the slider itself just use the setter to set the thing

Class Slider - template M_TYPE: public Mouse Input Handler Observer Class
Constructor - posX, posY, Width, Height, FontSize
void update_Dropdown_menu
void draw _Dropdown_menu
void populate - M_TYPE new_option
void remove - M_TYPE target
M_TYPE select_option
virtual void updateMousePos - int posX, int posY = 0
virtual void updateMouseClicked - key KeyPressed
Destructor
int pos_x, pos_y, width, height, font_size



