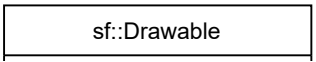
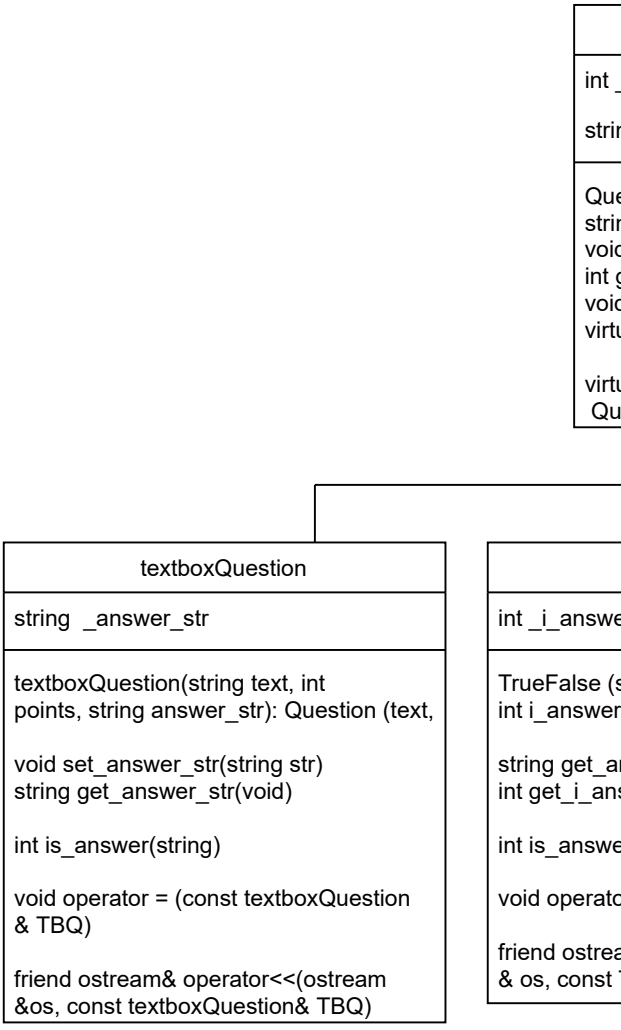
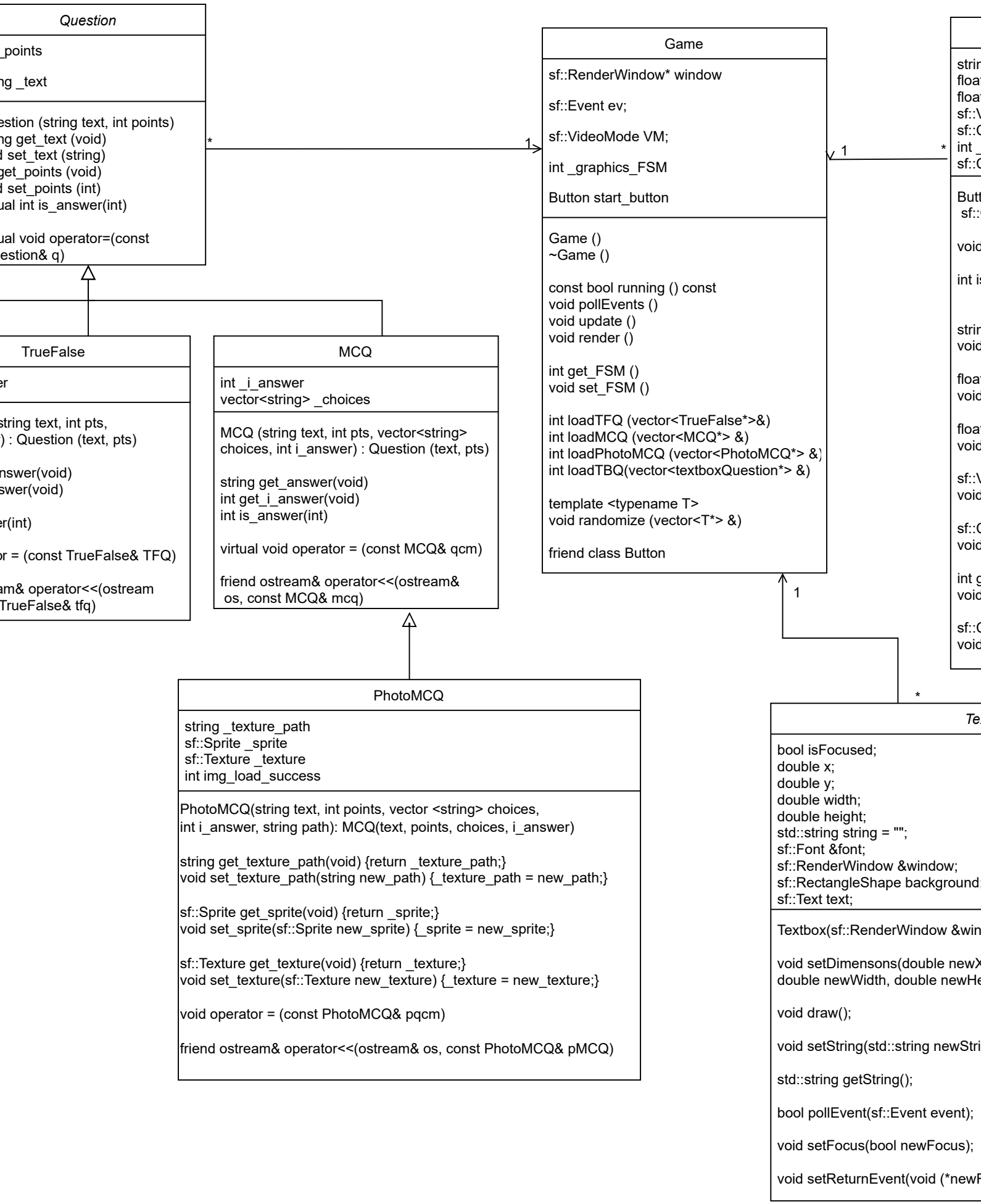
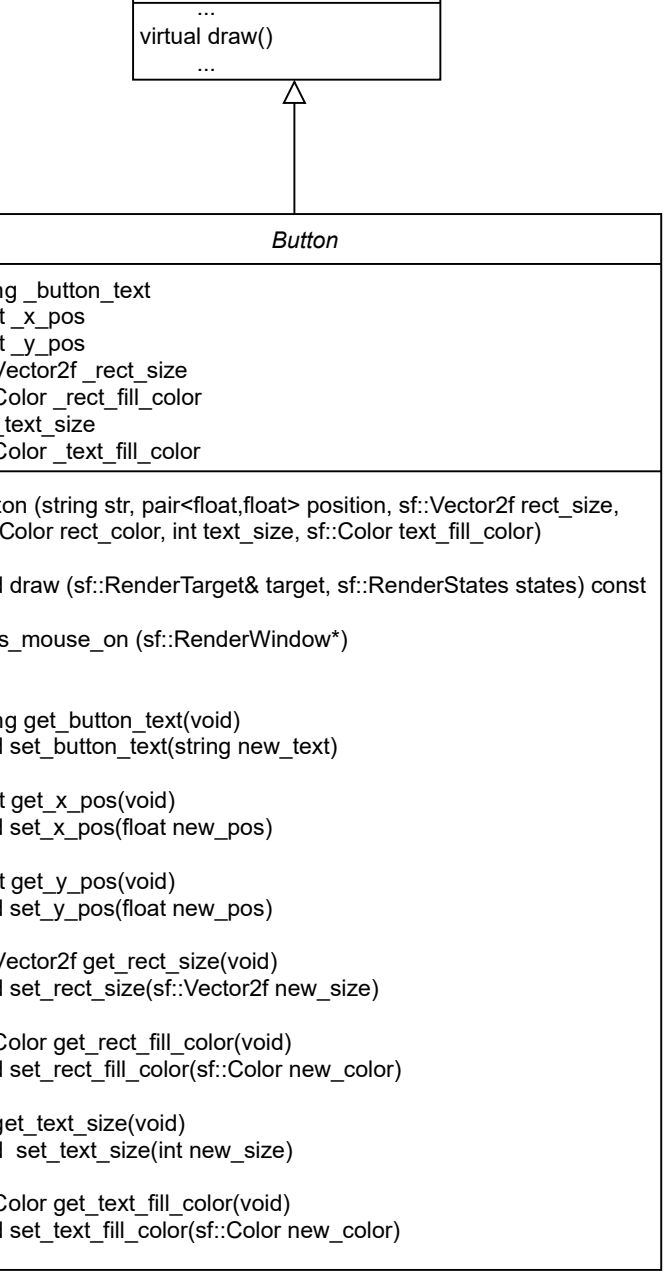

sf::Drawable


int _
stri
Que
stri
voi
int g
voi
virtu
virtu
Qu
textboxQuestion
string _answer_str
textboxQuestion(string text, int points, string answer_str): Question (text,
void set_answer_str(string str)
string get_answer_str(void)
int is_answer(string)
void operator = (const textboxQuestion & TBQ)
friend ostream& operator<<(ostream &os, const textboxQuestion& TBQ)
int _i_answe
TrueFalse (s
int i_answer
string get_a
int get_i_ans
int is_answe
void operato
friend ostrea
& os, const


Question
_points
ng _text
estion (string text, int points)
ng get_text (void)
d set_text (string)
get_points (void)
d set_points (int)
ual int is_answer(int)
ual void operator=(const
estion& q)
*
1
Game
sf::RenderWindow* window
sf::Event ev;
sf::VideoMode VM;
int _graphics_FSM
Button start_button
Game ()
~Game ()
const bool running () const
void pollEvents ()
void update ()
void render ()
int get_FSM ()
void set_FSM ()
int loadTFQ (vector<TrueFalse*>&)
int loadMCQ (vector<MCQ*> &)
int loadPhotoMCQ (vector<PhotoMCQ*> &)
int loadTBQ(vector<textboxQuestion*> &)
template <typename T>
void randomize (vector<T*> &)
friend class Button
1
*
TrueFalse
er
string text, int pts,
) : Question (text, pts)
nswer(void)
swer(void)
er(int)
or = (const TrueFalse& TFQ)
am& operator<<(ostream
TrueFalse& tfq)
MCQ
int _i_answer
vector<string> _choices
MCQ (string text, int pts, vector<string>
choices, int i_answer) : Question (text, pts)
string get_answer(void)
int get_i_answer(void)
int is_answer(int)
virtual void operator = (const MCQ& qcm)
friend ostream& operator<<(ostream&
os, const MCQ& mcq)
PhotoMCQ
string _texture_path
sf::Sprite _sprite
sf::Texture _texture
int img_load_success
PhotoMCQ(string text, int points, vector <string> choices,
int i_answer, string path): MCQ(text, points, choices, i_answer)
string get_texture_path(void) {return _texture_path;}
void set_texture_path(string new_path) {_texture_path = new_path;}
sf::Sprite get_sprite(void) {return _sprite;}
void set_sprite(sf::Sprite new_sprite) {_sprite = new_sprite;}
sf::Texture get_texture(void) {return _texture;}
void set_texture(sf::Texture new_texture) {_texture = new_texture;}
void operator = (const PhotoMCQ& pqcm)
friend ostream& operator<<(ostream& os, const PhotoMCQ& pMCQ)
*
bool isFocused;
double x;
double y;
double width;
double height;
std::string string = "";
sf::Font &font;
sf::RenderWindow &window;
sf::RectangleShape background;
sf::Text text;
Textbox(sf::RenderWindow &win
void setDimensons(double newX
double newWidth, double newHe
void draw();
void setString(std::string newStri
std::string getString();
bool pollEvent(sf::Event event);
void setFocus(bool newFocus);
void setReturnEvent(void (*newF


...
virtual draw()
...
Button
g _button_text
t _x_pos
t _y_pos
Vector2f _rect_size
Color _rect_fill_color
_text_size
Color _text_fill_color
on (string str, pair<float,float> position, sf::Vector2f rect_size,
Color rect_color, int text_size, sf::Color text_fill_color)
l draw (sf::RenderTarget& target, sf::RenderStates states) const
s_mouse_on (sf::RenderWindow*)
g get_button_text(void)
l set_button_text(string new_text)
t get_x_pos(void)
l set_x_pos(float new_pos)
t get_y_pos(void)
l set_y_pos(float new_pos)
Vector2f get_rect_size(void)
l set_rect_size(sf::Vector2f new_size)
Color get_rect_fill_color(void)
l set_rect_fill_color(sf::Color new_color)
get_text_size(void)
l set_text_size(int new_size)
Color get_text_fill_color(void)
l set_text_fill_color(sf::Color new_color)
xtbox
;
dowToUse, sf::Font &fontToUse);
X, double newY,
eight);
ng);
FunctionToCall()));