```
#ifndef BOTTLE_H
#define BOTTLE_H
#endif
#ifndef ABSTRACTTEXTURE_H
#define ABSTRACTTEXTURE_H
#include "AbstractTexture.h"
#endif
#ifndef OBSERVER_H
#define OBSERVER_H
#include "Observer.h"
#endif
#ifndef LEVELELEMENTINTERFACE_H
#define LEVELELEMENTINTERFACE_H
#include "LevelElementInterface.h"
#endif
#include "SDL_mixer.h"

class Bottle : public AbstractTexture, public Observer, public LevelElementInterface{
    private:
        SDL_Texture* bottle_tx;
        SDL_Texture* bottle_shattered_tx;
        SDL_Texture* curr_tx;
        int ticks_to_self_destruction;
        bool headLeft;
        int harmful;
        int sinus_ticks;
        bool reached_max_height;
        double turn;
        int ground_y_coordinate;
    public:
        Bottle(int x, int y, bool headLeft);
        bool shatter();
        void renderCollider();
        ~Bottle();
        //Abstract Texture
        bool loadMedia();
        //LevelElementInterface
        virtual void render();
        virtual void tick();
        virtual void restart();
        virtual void close();
        virtual std::string getType(); //"PLAYERRIGHT", "PLAYERLEFT", "BOTTLE", "BACKGROUND"
        Bottle* spawnBottle();
        virtual void checkInput();
        virtual std::string isDead(); //"NOTDEAD", "DEADPLAYER", "BROKENBOTTLE"
        //Observer
        void update(int collided_with, int own_collider, SDL_Rect rec);
        int getColliderType(int index_in_vector);
        std::vector<SDL_Rect> getColliders();
};
```

51
52
53