```
#ifndef PLAYER_H
#define PLAYER_H
#endif
#ifndef ABSTRACTTEXTURE_H
#define ABSTRACTTEXTURE_H
#include "AbstractTexture.h"
#endif
#ifndef HEALTHBAR_H
#define HEALTHBAR_H
#include "Healthbar.h"
#endif
#ifndef BOTTLE_H
#define BOTTLE_H
#include "Bottle.h"
#endif
#ifndef LEVELELEMENTINTERFACE_H
#define LEVELELEMENTINTERFACE_H
#include "LevelElementInterface.h"
#endif
#include "SDL.h"
#include <string>
#include <array>
#include <stack>

class Player : public AbstractTexture, public Observer, public LevelElementInterface{
  protected:
      std::array<SDL_Texture*, 7> spritesheets;
      std::array<int, 2> curr_state;  //{current picture of state, current state}
      std::array<int, 7> max_sprites;
      std::stack<int> height_stack;
      int heightAboveTheGround;
      Healthbar* healthbar = NULL;
      int ticked;
      bool headLeft;
      bool blocking; //refers to the state, if the Player is blocking at the moment, this is true
      std::string collision_direction; //can be "NONE", "LEFT" or "RIGHT"
      void renderUnflipped();
      void renderFlipped();
      void renderColliders();
      void changeStateTo(int state);
  public:
      enum states { IDLE, BLOCK, WALK, HURT, JUMP, STAB, THROWBOTTLE };
      enum collider_types { BODY, SWORD, BOTTLE, BOTTLEHARMLESS };
      Player(bool headLeft);
      ~Player();
      void moveX(bool caused_by_collision, int distance);
      int getState();
      //Abstract Texture
      bool loadMedia();
      //Observer Interface
```

```
51          void update(int collided_with, int own_collider, SDL_Rect rec);
52          int getColliderType(int index_in_vector);
53          std::vector<SDL_Rect> getColliders(); //[BODY] for the body collider [SWORD] for the sword colliders
54          //LevelElementInterface
55          virtual void close();
56          virtual std::string isDead();
57          virtual void render();
58          virtual void checkInput();
59          virtual void restart();
60          virtual void tick();
61          virtual std::string getType(); //"PLAYERRIGHT", "PLAYERLEFT", "BOTTLE", "BACKGROUND", "PLAYER" (from player class)
62          Bottle* spawnBottle();
63  };
64
```