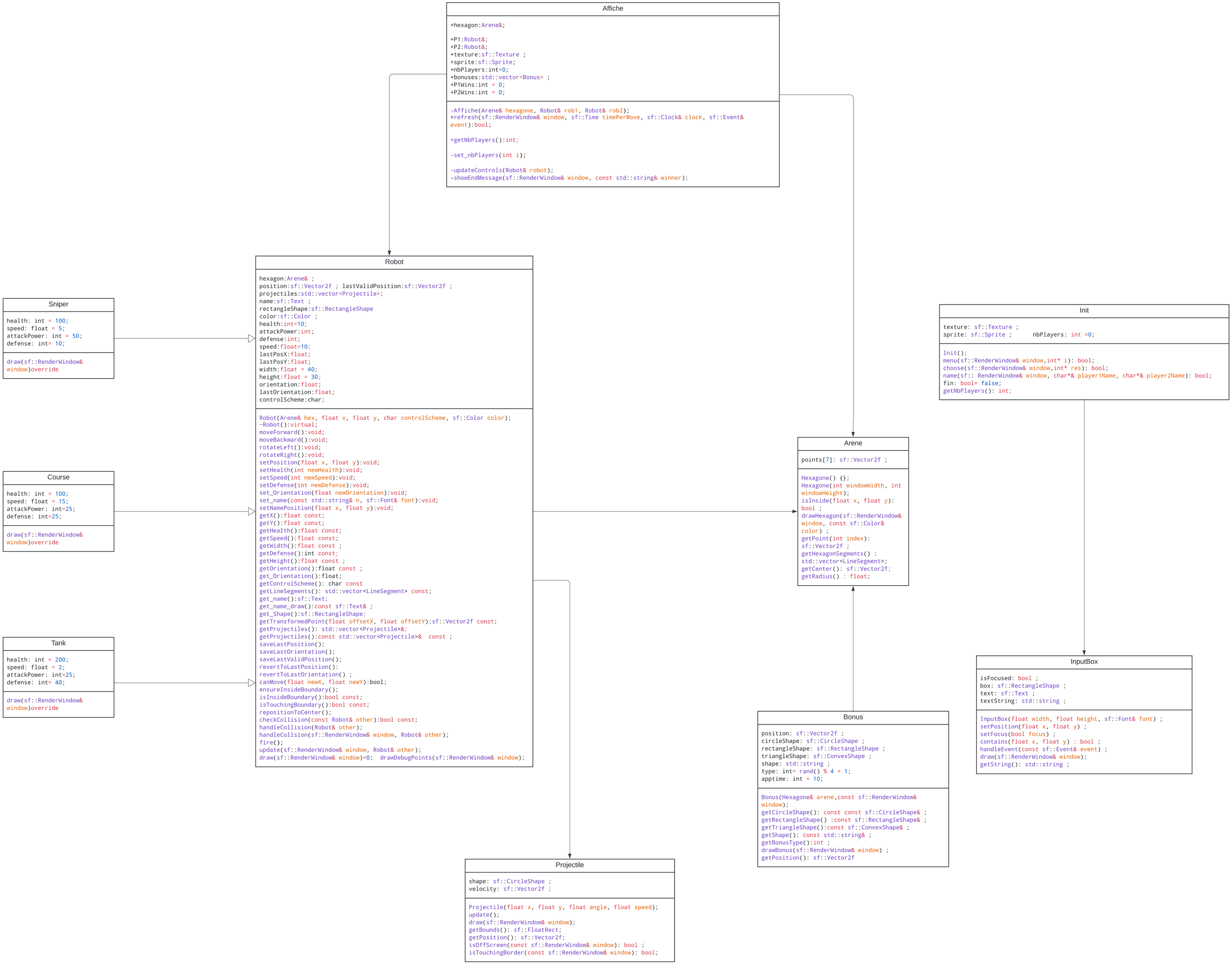
Affiche
+hexagon:Arene&;
+P1:Robot&;
+P2:Robot&;
+texture:sf::Texture ;
+sprite:sf::Sprite;
+nbPlayers:int=0;
+bonuses:std::vector<Bonus> ;
+P1Wins:int = 0;
+P2Wins:int = 0;
-Affiche(Arene& hexagone, Robot& rob1, Robot& rob2);
+refresh(sf::RenderWindow& window, sf::Time timePerMove, sf::Clock& clock, sf::Event&
event):bool;
+getNbPlayers():int;
-set_nbPlayers(int i);
-updateControls(Robot& robot);
-showEndMessage(sf::RenderWindow& window, const std::string& winner);
Robot
hexagon:Arene& ;
position:sf::Vector2f ; lastValidPosition:sf::Vector2f ;
projectiles:std::vector<Projectile>;
name:sf::Text ;
rectangleShape:sf::RectangleShape
color:sf::Color ;
health:int=10;
attackPower:int;
defense:int;
speed:float=10;
lastPosX:float;
lastPosY:float;
width:float = 40;
height:float = 30;
orientation:float;
lastOrientation:float;
controlScheme:char;
Robot(Arene& hex, float x, float y, char controlScheme, sf::Color color);
~Robot():virtual;
moveForward():void;
moveBackward():void;
rotateLeft():void;
rotateRight():void;
setPosition(float x, float y):void;
setHealth(int newHealth):void;
setSpeed(int newSpeed):void;
setDefense(int newDefense):void;
set_Orientation(float newOrientation):void;
set_name(const std::string& n, sf::Font& font):void;
setNamePosition(float x, float y):void;
getX():float const;
getY():float const;
getHealth():float const;
getSpeed():float const;
getWidth():float const ;
getDefense():int const;
getHeight():float const ;
getOrientation():float const ;
get_Orientation():float;
getControlScheme(): char const
getLineSegments(): std::vector<LineSegment> const;
get_name():sf::Text;
get_name_draw():const sf::Text& ;
get_Shape():sf::RectangleShape;
getTransformedPoint(float offsetX, float offsetY):sf::Vector2f const;
getProjectiles(): std::vector<Projectile>&;
getProjectiles():const std::vector<Projectile>& const ;
saveLastPosition();
saveLastOrientation();
saveLastValidPosition();
revertToLastPosition():
revertToLastOrientation() ;
canMove(float newX, float newY):bool;
ensureInsideBoundary();
isInsideBoundary():bool const;
isTouchingBoundary():bool const;
repositionToCenter();
checkCollision(const Robot& other):bool const;
handleCollision(Robot& other);
handleCollsion(sf::RenderWindow& window, Robot& other);
fire();
update(sf::RenderWindow& window, Robot& other);
draw(sf::RenderWindow& window)=0; drawDebugPoints(sf::RenderWindow& window);
Sniper
health: int = 100;
speed: float = 5;
attackPower: int = 50;
defense: int= 10;
draw(sf::RenderWindow&
window)override
Course
health: int = 100;
speed: float = 15;
attackPower: int=25;
defense: int=25;
draw(sf::RenderWindow&
window)override
Tank
health: int = 200;
speed: float = 2;
attackPower: int=25;
defense: int= 40;
draw(sf::RenderWindow&
window)override
Projectile
shape: sf::CircleShape ;
velocity: sf::Vector2f ;
Projectile(float x, float y, float angle, float speed);
update();
draw(sf::RenderWindow& window);
getBounds(): sf::FloatRect;
getPosition(): sf::Vector2f;
isOffScreen(const sf::RenderWindow& window): bool ;
isTouchingBorder(const sf::RenderWindow& window): bool;
Arene
points[7]: sf::Vector2f ;
Hexagone() {};
Hexagone(int windowWidth, int
windowHeight);
isInside(float x, float y):
bool ;
drawHexagon(sf::RenderWindow&
window, const sf::Color&
color) ;
getPoint(int index):
sf::Vector2f ;
getHexagonSegments() :
std::vector<LineSegment>;
getCenter(): sf::Vector2f;
getRadius() : float;
Bonus
position: sf::Vector2f ;
circleShape: sf::CircleShape ;
rectangleShape: sf::RectangleShape ;
triangleShape: sf::ConvexShape ;
shape: std::string ;
type: int= rand() % 4 + 1;
apptime: int = 10;
Bonus(Hexagone& arene,const sf::RenderWindow&
window);
getCircleShape(): const const sf::CircleShape& ;
getRectangleShape() :const sf::RectangleShape& ;
getTriangleShape():const sf::ConvexShape& ;
getShape(): const std::string& ;
getBonusType():int ;
drawBonus(sf::RenderWindow& window) ;
getPosition(): sf::Vector2f
Init
texture: sf::Texture ;
sprite: sf::Sprite ; nbPlayers: int =0;
Init();
menu(sf::RenderWindow& window,int* i): bool;
choose(sf::RenderWindow& window,int* res): bool;
name(sf:: RenderWindow& window, char*& player1Name, char*& player2Name): bool;
fin: bool= false;
getNbPlayers(): int;
InputBox
isFocused: bool ;
box: sf::RectangleShape ;
text: sf::Text ;
textString: std::string ;
InputBox(float width, float height, sf::Font& font) ;
setPosition(float x, float y) ;
setFocus(bool focus) ;
contains(float x, float y) : bool ;
handleEvent(const sf::Event& event) ;
draw(sf::RenderWindow& window);
getString(): std::string ;