**EnemyProjectile (pointer to Player, PlayArea)**

**Aggregation to Enemy**

+ GetEnemyDamage() : int

+ GetEnemyProjectileBoundingBox() : FloatRect

+ setPosition(Vector2f) : void

+ Intersect() : bool

+ Update() : void

+ Draw(RenderWindow &window) : void

- enemyProjectileText : Texture

- enemyProjectileSprite : Sprite

- enemyProjectileSpeed : float

- enemyProjectileNormalSpeed : float

- enemyProjectileDamage : int

- <Enemy\*> enemies : vector

**Enemy (pointer to Player, PlayArea)**

+ GetEnemyBoundingBox() : FloatRect

+ SetPosition() : void

+ GetPosition() : Vector2f

+ Intersect() : bool

+ GetEnemyHP() : int

+ GetEnemyDamage() : int

+ TakeDamage(int damage) : void

+ SetSprite() : void

+ IsDead() : bool

+ Update() : void

+ Draw(RenderWindow &window) : void

- enemyText : Texture

- enemyHurtText : Texture

- enemySprite : Sprite

- enemyBoundingBox : FloatRect

- enemySpeed : float

- enemyNormalSpeed : float

- enemyHp : int

- enemyDMG : int

**BackGround**

+ Update() : void

+ Draw(RenderWindow &window) : void

- bg1Tex : Texture

- bg2Tex : Texture

- bgSize : Vector2f

- bg1Sprite : Sprite

- bg2Sprite : Sprite

- bgSpeed : float

- bgY : float

- bg1Y : float

- bg2Y : float

- windowBgDiff : float

**StarField**

+ Update() : void

+ Draw(RenderWindow &window) : void

- star1Tex : Texture

- star2Tex : Texture

- star1Sprite : Sprite

- star2Sprite : Sprite

- starSpeed : float

- sfY : float

- sf1Y : float

- sf2Y : float

- windowBgDiff : float

**ElapsedTime (pointer to Menu)**

**+ font : Font**

**+ text : Text**

**+ clock : Clock**

**+ playtime : Time**

**+ getElapsedTime() : int**

**+ Update() : void**

**+ Draw(RenderWindow &window) : void**

**- mClockReset : bool**

**ScoreText**

**+ getScore() : int**

**+ Update() : void**

**+ Draw(RenderWindow &window) : void**

**- font : Font**

**- text : Text**

**- score : int**

**HealthPack (pointer to PlayArea, Player)**

+ GetHealthPackBoundingBox() : FloatRect

+ SetPosition() : void

+ Intersect() : bool

+ GetHealth() : int

+ Update() : void

+ Draw(RenderWindow &window) : void

- healthText : Texture

- healthSprite : Sprite

- healthBoundingBox : FloatRect

- healthSpeed : float

- healthNormalSpeed : float

- health : int

**SoundManager**

+ EnemyDeathSound() : void

+ ProjectileSound() : void

+ PlayerDeathSound() : void

+ HealthPickupSound() : void

+ PlayerHurt() : void

+ GameOver() : void

- enemyDeath : SoundBuffer

- projectileSound : SoundBuffer

- playerDeath : SoundBuffer

- playerHurt : SoundBuffer

- healthPickUp : SoundBuffer

- gameOver : SoundBuffer

- enemydeath : Sound

- projectilesound : Sound

- playerdeath : Sound

- playerhurt : Sound

- healthpickup : Sound

- gameover : Sound

**Game (pointer to Player, PlayArea, BackGround, StarField, ScoreText, ElapsedTime, SoundManager, Menu)**

**Aggregation to Projectile, Enemy, EnemyProjectile, Asteroid, Healthpack**

**+ UpdateSpawnTimer() : void**

**+ Update() : void**

**+ Draw() : void**

**- <Projectile\*> projectiles : vector**

**- <Enemy\*> enemies : vector**

**- <EnemyProjectile\*> enemyprojectiles : vector**

**- <HealthPack\*> healthpacks : vector**

**- event : Event**

**- projectileTimer : float**

**- healthTimer : float**

**- spawnTimer : float**

**- asteroidTimer : float**

**- spawnTimerValue : float**

**- lastSpawnTimerValue : int**

**- spawnRateTime : int**

**- asteroidTimerFocus : float**

**- asteroidTimerNormal : float**

**- asteroidTimerValue : float**

**- lastAsteroidUpdate : int**

**- asteroidTime : int**

**- healthTimerValue : float**

**- lastHealthTimerUpdate : int**

**- healthTime : int**

**- healthTop : int**

**- healthBottom : int**

**- windowWidth : int**

**- windowHeight : int**

**- score : int**

**- isPlaying : bool**

**In loving memory of Leena Heikkinen (RIP)**

**Menu (pointer to ScoreText)**

**+ IsPlaying() : bool**

**+ isNotPlaying() : bool**

**+ stopPlaying() : void**

**+ startPlaying() : void**

**+ Draw(RenderdWindow &window) : void**

**+ DrawEndMenu(RenderWindow &window) : void**

**+ Update() : void**

**+ isPlaying : bool**

**- font : Font**

**- title : Text**

**- start : Text**

**- quit : Text**

**- controls : Text**

**- move : Text**

**- shoot : Text**

**- focus : Text**

**- dead : Text**

**- score : Text**

**Asteroid (pointer to PlayArea, Player)**

+ GetAsteroidBoundingBox() : FloatRect

+ SetPosition() : void

+ Intersect() : bool

+ GetAsteroidHP() : int

+ GetAsteroidDamage() : int

+ TakeDamage(int damage) : void

+ SetSprite() : void

+ IsDestroyed() : bool

+ Update() : void

+ Draw(RenderWindow& window) : void

- asteroidText : Texture

- asteroidHurtText : Texture

- asteroidSprite : Sprite

- asteroidBoundingBox : FloatRect

- asteroidSpeed : float

- asteroidHP : int

- asteroidDMG : int

**Projectile (pointer to Player, PlayArea)**

+ GetProjectileDamage() : int

+ GetProjectileBoundingBox() : FloatRect

+ SetPosition(Vector2f) : void

+ Intersect() : bool

+ Update() : void

+ Draw(RenderWindow &window) : void

- projectileText : Texture

- projectileSprite : Sprite

- projectileSpeed : float

- projectileNormalSpeed : float

- projectileDamage : int

**Player (pointer to PlayArea)**

+ GetPosition() : Vector2f

+ GetPlayerBoundingBox() : FloatRect

+ SetPosition() : void

+ GetPlayerHP() : int

+ TakeDamage(int damage) : void

+ TakeHealth(int health) : void

+ IsDead() : bool

+ Update() : void

+ PlayerInput() : void

+ Draw(RenderWindow &window) : void

+ PlayerFocus() : bool

+ GetPlayerFocusFactor() : float

+ font : Font

+ text : Text

+ focus : Text

- playerText : Texture

- playerHurtText : Texture

- playerPickupText : Texture

- playerSprite : Sprite

- playerBoundingBox : FloatRect

- playerSpeed : float

- playerNormalSpeed : float

- playerFocusFactor : float

- playerHP : int

- playerFullHP : int

**PlayArea**

+ GetTopBoundingBox() : FloatRect

+ GetBottomBoundingBox() : FloatRect

+ GetRightBoundingBox() : FloatRect

+ GetLeftBoundingBox() : FloatRect

+ SetPositions() : void

+ Draw(RenderWindow &window) : void

- topBorder : RectangleShape

- bottomBorder : RectangleShape

- rightBorder : RectangleShape

- leftBorder : RectangleShape

- topBoundingBox : FloatRect

- bottomBoundingBox : FloatRect

- rightBoundingBox : FloatRect

- width : int

- height : int

- borderSize : int