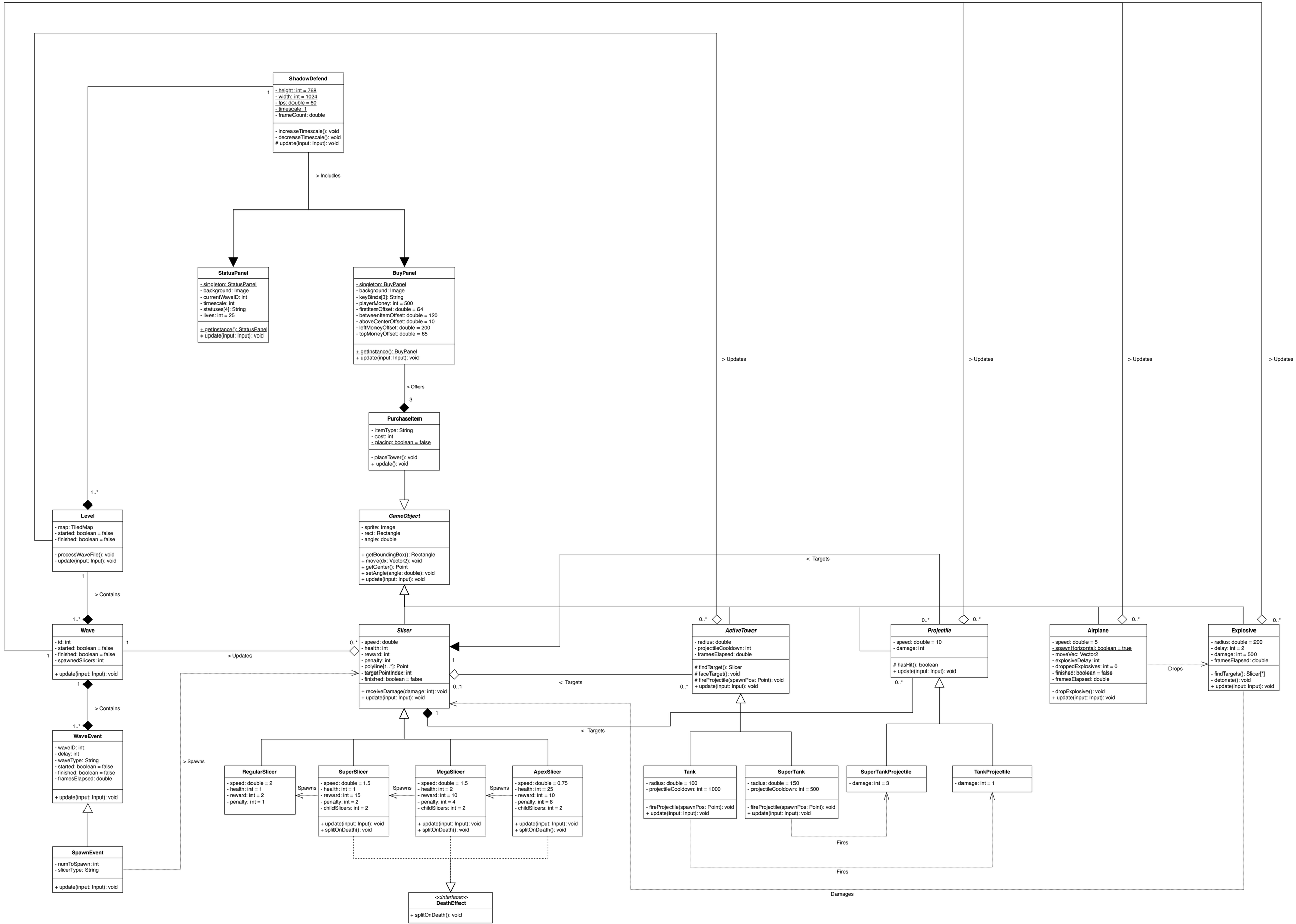
ShadowDefend
- height: int = 768
- width: int = 1024
- fps: double = 60
- timescale: 1
- frameCount: double
- increaseTimescale(): void
- decreaseTimescale(): void
update(input: Input): void
> Includes
StatusPanel
- singleton: StatusPanel
- background: Image
- currentWaveID: int
- timescale: int
- statuses[4]: String
- lives: int = 25
+ getInstance(): StatusPanel
+ update(input: Input): void
BuyPanel
- singleton: BuyPanel
- background: Image
- keyBinds[3]: String
- playerMoney: int = 500
- firstItemOffset: double = 64
- betweenItemOffset: double = 120
- aboveCenterOffset: double = 10
- leftMoneyOffset: double = 200
- topMoneyOffset: double = 65
+ getInstance(): BuyPanel
+ update(input: Input): void
> Offers
PurchaseItem
- itemType: String
- cost: int
- placing: boolean = false
- placeTower(): void
+ update(): void
GameObject
- sprite: Image
- rect: Rectangle
- angle: double
+ getBoundingBox(): Rectangle
+ move(dx: Vector2): void
+ getCenter(): Point
+ setAngle(angle: double): void
+ update(input: Input): void
Level
- map: TiledMap
- started: boolean = false
- finished: boolean = false
- processWaveFile(): void
- update(input: Input): void
> Contains
Wave
- id: int
- started: boolean = false
- finished: boolean = false
- spawnedSlicers: int
+ update(input: Input): void
> Updates
> Contains
WaveEvent
- waveID: int
- delay: int
- waveType: String
- started: boolean = false
- finished: boolean = false
- framesElapsed: double
+ update(input: Input): void
SpawnEvent
- numToSpawn: int
- slicerType: String
+ update(input: Input): void
> Spawns
Slicer
- speed: double
- health: int
- reward: int
- penalty: int
- polyline[1..*]: Point
- targetPointIndex: int
- finished: boolean = false
+ receiveDamage(damage: int): void
+ update(input: Input): void
< Targets
RegularSlicer
- speed: double = 2
- health: int = 1
- reward: int = 2
- penalty: int = 1
Spawns
SuperSlicer
- speed: double = 1.5
- health: int = 1
- reward: int = 15
- penalty: int = 2
- childSlicers: int = 2
+ update(input: Input): void
+ splitOnDeath(): void
Spawns
MegaSlicer
- speed: double = 1.5
- health: int = 2
- reward: int = 10
- penalty: int = 4
- childSlicers: int = 2
+ update(input: Input): void
+ splitOnDeath(): void
Spawns
ApexSlicer
- speed: double = 0.75
- health: int = 25
- reward: int = 10
- penalty: int = 8
- childSlicers: int = 2
+ update(input: Input): void
+ splitOnDeath(): void
<<Interface>>
DeathEffect
+ splitOnDeath(): void
ActiveTower
- radius: double
- projectileCooldown: int
- framesElapsed: double
findTarget(): Slicer
faceTarget(): void
fireProjectile(spawnPos: Point): void
+ update(input: Input): void
Projectile
- speed: double = 10
- damage: int
hasHit(): boolean
+ update(input: Input): void
Airplane
- speed: double = 5
- spawnHorizontal: boolean = true
- moveVec: Vector2
- explosiveDelay: int
- droppedExplosives: int = 0
- finished: boolean = false
- framesElapsed: double
- dropExplosive(): void
+ update(input: Input): void
Drops
Explosive
- radius: double = 200
- delay: int = 2
- damage: int = 500
- framesElapsed: double
- findTargets(): Slicer[*]
- detonate(): void
+ update(input: Input): void
Tank
- radius: double = 100
- projectileCooldown: int = 1000
- fireProjectile(spawnPos: Point): void
+ update(input: Input): void
SuperTank
- radius: double = 150
- projectileCooldown: int = 500
- fireProjectile(spawnPos: Point): void
+ update(input: Input): void
SuperTankProjectile
- damage: int = 3
TankProjectile
- damage: int = 1
Fires
Fires
Damages
1
1..*
1..*
1
1..*
1
1
1
1
0..*
0..1
0..*
1
0..*
0..*
0..*
0..*
0..*
0..*
0..*
0..*
0..*
0..*