UML class diagram.

ShadowDefend
- HEIGHT: int = 768
- WIDTH: int = 1024
- MAX_LEVELS_SUPPORTED: int = 2
- FPS: int = 60
- levelNumber: int = 1

+ main(args: String[]): void
# update(input: Input): void

ie. new Level overwrites old in ShadowDefend

Level
- WAVES_SRC: String = "res/levels/waves.txt"
- LEVELS_PATH: String = "res/levels"
- INDEX_OF_WAVE_NUM: int = 0
- NEW_WAVE_KEY: Keys = Keys.S
- SPEED_UP_KEY: Keys = Keys.L
- SLOW_DOWN_KEY: Keys = Keys.K
- SENSITIVITY_FOR_KEY: double = 0.5
- INITIAL_NUM_LIVES: int = 25
- INITIAL_MONEY: int = 500
- WAVE_STATUS: String = "Wave in Progress"
- PLACING_STATUS: String = "Placing"
- WINNER_STATUS: String = "Winner!"
- WAITING_STATUS: String = "Awaiting Start"
- frameCount: double = 0;
- frameOfLatestChanger: double = 0
- timescale: int = 1
- livesLeft: int = INITIAL_NUM_LIVES
- moneyLeft: int = INITIAL_MONEY
- map: TiledMap
- wavesCompleted: int = 0
- isActive: boolean = true
- gameIsOver: boolean = false
- isPlacing: boolean = false

+ Level(levelNumber: int)
- adjustTimescale(adjustment: int, frameCount: double): void
+ update(input: Input): void

Wave
- INDEX_OF_WAVE_TYPE: int = 1
- SPAWN_EVENT_TYPE: String = "spawn"
- DELAY_EVENT_TYPE: String = "delay"
- MONEY_AWARDED: int = 150
- LEVEL_NUM_BONUS: int = 100
- polyline: ArrayList<Point>
- isActive: boolean = false
- eventsCommenced: int = 0
- waveNumber: int

+ Wave(polyline: List<Point>, waveNumber: int)
+ addWaveEvent(waveEventInfo: String): void
+ update(timescale: int, frameCount: double): void

WaveEvent *(italic)*
- INDEX_OF_WAVE_NUM: int = 0
- MILLI_TO_NORMAL: double = 0.001
- waveNumber: int
- isActive: boolean = false
- frameOfEventStart: double

WaveEvent(eventInfo: String[])
+ activate(frameCount: double): void
+ isStillRunning(): boolean
+ update(timescale: int, frameCount: double): void

SpawnEvent
- INDEX_OF_NUM_SPAWN: int = 2
- INDEX_OF_ENEMY_TYPE: int = 3
- INDEX_OF_SPAWN_DELAY: int = 4
- ONE_PIXEL: int = 1
- REG_SLICER_NAME: String = "slicer"
- SUPER_SLICER_NAME: String = "superslicer"
- MEGA_SLICER_NAME: String = "megaslicer"
- APEX_SLICER_NAME: String = "apexslicer"
- numSpawned: int = 0
- numDeactivated: int = 0
- numToSpawn: int
- spawnDelay: int
- frameOfLatestChange: double
- stillRunning: boolean
- polyline: ArrayList<Point>

+ SpawnEvent(eventInfo: String[], polyline: ArrayList<Point>)
<<override>> + activate(frameCount: double): void
<<override>> + isStillRunning(): boolean
<<override>> + update(timescale: int, framecount: double): void

DelayEvent
- INDEX_OF_DELAY_LENGTH: int = 2
- delayLength: int

+ DelayEvent(eventInfo: String[])
+ <<override>> isStillRunning(): boolean
+ <<override>> update(timescale: int, frameCount: double): void

Defender *(italic)*
- image: Image
- rect: Rectangle
- damage: double
- price: int
- position: Point
- isDeployed: boolean = false

Defender(point: Point, image: Image)
+ purchase(): int

Tower *(italic)*
- MILLI_TO_NORMAL: double = 0.001
- radius: int
- coolDownTime: int
- framesSincePrevShot: double

Tower(point: Point, image: Image)
+ shoot(): void

Tank
- TANK_PRICE: int = 250
- TANK_SRC: String = "res/images/tank.png"
- RADIUS: int = 100
- COOLDOWN: int = 1000

+ Tank(point: Point)

SuperTank
- STANK_PRICE: int = 600
- STANK_SRC: String = "res/images/supertank.png"
- RADIUS: int = 150
- COOLDOWN: int = 500

+ SuperTank(point: Point)

Airplane
- AIRPLANE_PRICE: int = 500
- SPEED: int = 5
- AIRPLANE_SRC: String = "res/images/airsupport.png"
- isHorizontal: boolean
- waitTime: int

+ Airplane(point: Point)
+ deploy(point: Point): void
+ drop(point: Point): void

Panel *(italic)*
- FONT_SRC: String = "res/fonts/DejavuSans-Bold.ttf"
- image: Image
- rect: Rectangle

Panel(point: Point, image: Image)

BuyPanel
- MONEY_FONT_SIZE: int = 48
- BINDS_FONT_SIZE: int = 14
- PRICE_FONT_SIZE: int = 22
- BUY_PANEL_IMG: Image = "res/images/buypanel.png"
- BUY_PANEL_CENTRE: Point = Point(ShadowDefend.getWidth()/2, BUY_PANEL_IMG.getHeight()/2)
- KEY_BINDS: String = "Key binds:\n\nS - Start Wave\nL - Increase Timescale\nK - Decrease Timescale"
- MONEY_TEXT_XVALUE = ShadowDefend.getWidth() - 200
- MONEY_TEXT_YVALUE = 65
- BINDS_TEXT_XVALUE = ShadowDefend.getWidth()/2 - 60
- BINDS_TEXT_YVALUE = 20
- DEFENDER_IMG_YVAL = BUY_PANEL_IMG.getHeight()/2-10
- TANK_XVAL = 64
- PXLS_BTW_IMGS = 120

+ update(moneyLeft: int): void
+ cursorOverDefender(cursorPos: Point): Defender

StatusPanel
- STATUS_PANEL_IMG: Image = "res/images/statuspanel.png"
- STATUS_PANEL_CENTRE: Point = Point(ShadowDefend.getWidth()/2, ShadowDefend.getheight() - STATUS_PANEL_IMG.getHeight()/2
- STATUS_FONT_SIZE: int = 15
- WAVE_NUM_TEXT_XVALUE: int = 10
- TSCALE_TEXT_XVALUE: int = 10
- STATUS_TEXT_XVALUE: int = 10
- LIVES_TEXT_XVALUE: int = 10
- ALL_TEXT_YVALUE: int = ShadowDefend.getHeight() - STATUS_PANEL_IMG.getHeight()/2 + 5

+ update(waveNumber: int, status: String, timescale: int, livesLeft: int): void

Enemy *(italic)*
- image: Image
- rect: Rectangle
- isActive: boolean
- polyline: List<Point>
- currentPoint: Point
- nextPoint: Point
- nextPointIndex: int
- angle: double
- healthRemaining: double

Enemy(point: Point, imageSrc: String)
+ move(dx: Vector2): void
+ findCentre(): Point
+ activate(): void
+ changeMovement(): void
+ die(): int

RegularSlicer
- REG_SLICER_SRC: String = "res/images/slicer.png"
- INITIAL_HEALTH: double = 1
- REWARD: int = 2
- PENALTY: int = 1
- SPEED: double = 2

<<override>> + die(): int

SuperSlicer
- SUPER_SLICER_SRC: String = "res/images/superslicer.png"
- NUM_CHILDREN : int = 2
- INITIAL_HEALTH: double = RegularSlicer.getInitialHealth()
- REWARD: int =15
- PENALTY: int = NUM_CHILDREN * RegularSlicer.getPenalty()
- SPEED: double = 0.75 * RegularSlicer.getSpeed()

<<override>> + die(): int

MegaSlicer
- MEGA_SLICER_SRC: String = "res/images/megaslicer.png"
- NUM_CHILDREN : int = 2
- INITIAL_HEALTH: double = 2 * SuperSlicer.getInitialHealth()
- REWARD: int =10
- PENALTY: int = NUM_CHILDREN * SuperSlicer.getPenalty()
- SPEED: double = SuperSlicer.getSpeed()

<<override>> + die(): int

ApexSlicer
- APEX_SLICER_SRC: String = "res/images/apexslicer.png"
- NUM_CHILDREN : int = 4
- INITIAL_HEALTH: double = 25 * RegularSlicer.getInitialHealth()
- REWARD: int =150
- PENALTY: int = NUM_CHILDREN * MegaSlicer.getPenalty()
- SPEED: double = 0.5 * MegaSlicer.getSpeed()

<<override>> + die(): int

Ammunition *(italic)*
- damage: int
- image: Image
- rect: Rectangle

+ hasHit(): boolean
+ inflictDamage(): void

Projectile
- TANK_PROJ_SRC: String = "res/images/tank_projectile.png"
- STANK_PROJ_SRC: String = "res/images/supertank_projectile.png"
- TANK_DAMAGE: int = 1
- STANK_DAMAGE: int = TANK_DAMAGE * 3
- SPEED: int = 10
- position: Point
- direction: Vector2

+ Projectile(point: Point)
<<override>> + hasHit(): boolean
<<override>> + inflictDamage(): void
+ move(): void

Explosive
- DAMAGE: int = 500
- EFFECT_RADIUS: int = 200
- DETONATION_TIME; int = 2
- EXPLOSIVE_IMG: Image = "res/images/explosive.png"
- poisition: Point
- droppedAt: int

+ hasLanded(frameCount: double): boolean
<<override>> + hasHit(): boolean
<<override>> + inflictDamage(): void