Wave Event Level waveNumber: int <<get>> levelNum: int <<get>> Notes: waveProgress: String <<get>>> ShadowDefend All abstract methods map: TiledMap <<qet>> in the parent class eventType: String level: int are inherited and path: List <<get>> time: int overridden in the child + ShadowDefend() classes timescale: int <<get>> WaveEvent(WaveData String) + main(args: String[]): void + Level(levelNum: int) 1..* updateWaveProgress(): void # update(input: Input) void; + newLevel(): void - spawnWave(): void 1..* - drawStatusPanel() + readWaveData(): void drawBuyPanel() ⊦ updateWave(path: List): void + startWave(): void Tank endWave(): void + updateWave(): void tankProjGraphic: Image + updateTimescale(timescale: int): + drawMap(): void tankGraphic: Image + increaseTimescale(): void radius: int = 100 0..* + decreaseTimescale(): void damage: int = 10..* AbstractTank cooldown: int = 1000 PlayerData radius: int Airplane + Tank() playerdata: PlayerData Spawn Event speed: int = 5 cooldown: int time: int - money: int <<get>> - damage: int spawnNumber: int Delay Event - life: int <<qet>> time: int enemyType: String count: int delayTime: int delayTime: int - pPos: Point · PlayerData() cooldown: int - DelayEvent(WaveData: String) SpawnEvent(WaveData: String) +getInstance(): PlayerData + AbstractTank() + launch(): void SuperTank + getPos(): Point +resetPlayerData(): void AbstractEnemy + updatePos(towards: ApexSlicer MegaSlicer superTGraphic: Image Point): void # setAttributes(radius: int, vPos: Vector2 <<get>> +loseLife(): void megaGraphic: Image apexGraphic: Image 1..* damage: int, cooldown: int) superProjGraphic: Image speed: double +addMoney(reward: int): void startHealth: int = 25 startHealth: int = 2 - cooldown: int = 500 + checkBounds(map Map): - currHealth: int <<get\set>> +loseMoney(cost: int): void reward: int = 150 reward: int = 10 - damage: int = 3 + render(): void penalty: int <<set>> - radius: int = 150 speed: double = 0.75 speed: double = 1.5 could also 0..* Singleton +renderIndicator(mousePos: potentially put defaultSpeed: double Should only have Point): void Explosive the timescale penalty: int = 16 penalty: int = 4SuperTank() one instance of a + place(input: Input): void here instead of angle: double player and should radius: int = 150 passing it be accessible + launch(): void - MegaSlicer(Point: point, speed through methods ApexSlicer(Point: point, speed: anywhere currentIndex: int cooldown: int = 2000 double) - damage: int = 500- AbstractEnemy(point: Point, speed: double) time: int 0..* # setAttributes(defaultSpeed: double, reward: int, penalty: Slicer SuperSlicer Projectile + detonate(): void int, health: int): void the health calculations - vPos: Vector 2 regGraphic: Image superGraphic: Image will be done in the + updatePos(towards:Point, + drop(): void super class, the health speed: double): void speed: int = 10 - startHealth: int = 1 startHealth: int = 1 values in the child + render(graphic: Image): classes are used for - damage: int void - reward: int = 2 instantiating reward: int = 15 + decreaseSpeed(): void - speed: double = 2 + updatePos(towards: Point): speed: double = 1.5 + increaseSpeed(): void - penalty: int = 1 penalty: int = 2+ targetPos(enemy: AbstractEnemy): void + recieveDamage(damage: If projectile int): void Slicer(Point: point, speed: SuperSlicer(Point: point, + render(projectile: Image): void speed is double) speed: double) + slicerPenalty(penalty: int): constant is it affected by the + damage(damage: int): void timescale? + slicerDeath(): void