1. Regular Single Target Tower (Homing)
2. Canon Tower (AOE)
3. Slow Tower (Multi Target, Low Fire Rate, % Chance of Slowing)
4. Freeze Tower (Single Target, Low Fire Rate, % Chance of Freezing)
5. Poison Tower (DoT, % Chance of Poisoning)

Upgrade not linear, play around with math functions.

Structure -> Scenery, Tower

Structure: position[][]

**tower.java**

VARIABLES

damage: int

level: int

buyCost: int

sellCost: int

upgradeCost: int

fireRate: double

range: double

element: String

position: int[]

NEW

chancePercent: double (chance to inflict special effect)

METHODS

public tower(int damage, int level, int buyCost, int sellCost,

int upgradeCost, double fireRate, double range, String element)

//Prints the properties

public abstract void properties()

public abstract int upgrade(int points)

public void setTower(int[] position)

//Returns points

public int buyAndSetTower(int points, int[] position)

public boolean getDistance(int[] critterPosition)

public abstract void shoot(boolean inRange, int critterHP, double critterSpeed, String critterElement,

int critterDefense)

//Returns points

public int sell(int points)

SETTERS AND GETTERS FOR EVERY VARIABLES

+tower(int, int, int, int, int, double, double, String)

+properties()

+upgrade(int)

+setTower(int[])

+buyAndSetTower(int, int[])

+getDistance(int[])

+shoot(boolean, int, double, String, int)

+sell(int)

+getDamage()

+setDamage(int)

+getLevel()

+setLevel(int)

+getBuyCost()

+setBuyCost(int)

+getSellCost()

+setSellCost(int)

+getUpgradeCost()

+setUpgradeCost(int)

+getFireRate()

+setFireRate(double)

+getRange()

+setRange(double)

+getElement()

+setElement(String)

+getPosition()

+setPosition(int[])

Movable -> Projectile (visual only)