Enemy Class Documentation

Table of Contents

- Global Summary/External Dependencies: Page 2

- Backing Fields/Properties: Page 2

- Constructors: Page 3

- External Methods: Page 4

- Internal Methods: Page 5

Global Summary: Enemy class defines an enemy in a console game that moves from side to side of the play area randomly.

External Dependencies: Player.cs, Coin.cs, GameArea.cs

Backing Fields/Properties

Private int \_enemyXPos (get/set (value cannot be negative)): The horizontal position of the enemy on screen.

Private int \_enemyYPos (get/set (value cannot be negative)): The vertical position of the enemy on screen.

Private Sides \_emergingSide (get/set): The side that the wave of the enemies will come out from.

Public default constant: DEFAULT\_EMERGING\_SIDE

Private char \_enemyChar (get/set): The ASCII character that represents the enemy.

Public default constant: DEFAULT\_ENEMY\_CHAR

Private ConsoleColor \_enemyForeColor (get/set): The foreground color of the character representing the enemy.

Public default constant: DEFAULT\_ENEMY\_FORECOLOR

Private ConsoleColor \_enemyBackColor (get/set): The background color of the character representing the enemy.

Public default constant: DEFAULT\_ENEMY\_BACKCOLOR

Public enum Sides { Top = 1, Right = 2, Bottom = 3, Left = 4}: Makes it easier to set the emerging side of the enemy wave.

Constructors

Default/Zero parameter constructor () summary: Sets EnemyXPos and EnemyYPos to zero, and sets the EmergingSide, EnemyChar, EnemyForeColor and EnemyBackColor to their public constant defaults.

Triple-Parameter/Advanced Editor constructor (char enemyChar\_, ConsoleColor enemyForeColor\_, ConsoleColor enemyBackColor\_) summary: Sets EnemyXPos and EnemyYPos to zero, sets the EmergingSide to its default value, and sets the EnemyChar, EnemyForeColor and EnemyBackColor to their respective parameters.

External Methods

Public static Enemy[] GetArrayOfEnemies(int numberOfEnemies, GameArea area) summary: Overload takes an integer representing the amount of enemies to be generated and the GameArea they will be used on. Creates a new Enemy array and initializes each element with its default constructor. Returns the array.

Public static Enemy[] GetArrayOfEnemies(int numberOfEnemies, GameArea area, char enemyChar, ConsoleColor enemyForeColor, ConsoleColor enemyBackColor) summary: Overload takes an integer representing the amount of enemies to be generated, the game area, an enemy character, and enemy foreground/background colors. Creates a new Enemy array and initializes each element with its Advanced Editor Constructor. Returns the array.

Public static void ChangeSide (Enemy[] enemies, GameArea area, Random rnd) summary: Takes an array of enemies, a game area and a Random class instance as arguments, and changes the emerging side of the enemies once its wave is complete by setting its initial position to a randomly generated side. Returns void.

Public static void MoveEnemies (Enemy[] enemies, GameArea area, Random rnd) summary: Takes an array of enemies, a game area and a Random class instance as arguments, and moves the enemies’ coordinates depending on their EmergingSide. Returns void.

Public static void Render(Enemy[] enemies, Player player, GameArea area, Coin[] coins, Random rnd) summary: Takes an array of enemies, a Player object instance, a GameArea instance, a Coin array and a Random class instance as arguments. Erases previous enemy icons after the enemies have moved and re-draws them as well as the coins so their icons do not disappear. Returns void.

Public static void RenderInitial (Enemy[] enemies) summary: Takes an array of enemies as an argument and draws every instance based on their X and Y coordinates. Is essentially a stripped back version of the Render method that only draws the enemies and does not erase any previous enemy icons or re-draw any coins. Returns void.

Internal Methods

Private static bool CheckIfAtEnd(Enemy[] enemies, GameArea area) summary: Takes an array of enemies and a GameArea instance as arguments, and utilizes a checksum to see if all of the enemies have reached their ending position for that specific wave. Returns true if the enemies are at the end of the wave, returns false if not.

Private static bool DeleteWaveAtEnd(Enemy[] enemies, GameArea area) summary: Takes an array of enemies and a GameArea instance as arguments, and erases all of the enemies once they’ve reached the end of their wave and replaces their icons with the screengrass of the GameArea instance passed to it. Returns void.