GameArea Class Documentation

Table of Contents

- Global Summary/External Dependencies: Page 2

- Backing Fields/Properties: Page 2-4

- Constructors: Page 5

- External Methods: Page 6

- Internal Methods: Page 7

Global Summary: GameArea class defines a working play area for a console game, complete with limit calculations and full cosmetic and dimensional customizability

External Dependencies: Player.cs, Coin.cs

Backing Fields/Properties

Private char \_screengrassChar (get/set): The ASCII character that is used for the background of the main play area.

Public constant default: DEFAULT\_SCREENGRASS\_CHAR

Private char \_borderChar (get/set): The ASCII character that is used to draw the border.

Public constant default: DEFAULT\_BORDER\_CHAR

Private ConsoleColor \_screengrassForeColor (get/set): The foreground color of ScreenGrassChar.

Public constant default: DEFAULT\_SCREENGRASS\_FORECOLOR

Private ConsoleColor \_screengrassBackColor (get/set): The background color of ScreenGrassChar.

Public constant default: DEFAULT\_SCREENGRASS\_BACKCOLOR

Private ConsoleColor \_borderForeColor (get/set): The foreground color of BorderChar.

Public constant default: DEFAULT\_BORDER\_FORECOLOR

Private ConsoleColor \_borderBackChar (get/set): The background color of BorderChar

Public constant default: DEFAULT\_BORDER\_BACKCOLOR

Private ConsoleColor \_moreLivesRemainingColor (get/set): Color of the display when the player has three or more lives remaining.

Public constant default: DEFAULT\_MORE\_LIVES\_REMAINING\_COLOR

Private ConsoleColor \_twoLivesRemainingColor (get/set): Color of the display when the player has two lives remaining.

Public constant default: DEFAULT\_TWO\_LIVES\_REMAINING\_COLOR

Private ConsoleColor \_oneLifeRemainingColor (get/set): Color of the display when the player has one life remaining.

Public constant default: DEFAULT\_ONE\_LIFE\_REMAINING\_COLOR

Private int \_width (get/set (value cannot be negative)): The width of the main play area NOT COUNTING THE BORDER SPACE.

Public constant default: DEFAULT\_WIDTH

Private int \_height (get/set (value cannot be negative)): The height of the main play area.

Public constant default: DEFAULT\_HEIGHT

Private int \_borderWidth (get/set (value cannot be negative)): The width of the blank areas flanking either side of the level.

Public constant default: DEFAULT\_BORDER\_WIDTH

Private int \_upLimit (get/set (value cannot be negative)): The lowest possible Y coord the player can collide with.

Private int \_downLimit (get/set (value cannot be negative)): The highest possible Y coord the player can collide with.

Private int \_leftLimit (get/set (value cannot be negative)): The lowest possible X coord the player can collide with.

Private int \_rightLimit (get/set (value cannot be negative)): The highest possible X coord the player can collide with.

Constructors

Default/Zero-parameter constructor () summary: ScreengrassChar, BorderChar, ScreengrassForeColor, ScreengrassBackColor, BorderForeColor, BorderBackColor, Width, Height, BorderWidth, MoreLivesRemainingColor, TwoLivesRemainingColor, OneLifeRemainingColor are all set to their public constant defaults. Limits are calculated (see CalculateLimits()).

Simple Editor/Dual-Parameter constructor (int width\_, int height\_) summary: ScreengrassChar, BorderChar, ScreengrassForeColor, ScreengrassBackColor, BorderForeColor, BorderBackColor, BorderWidth, MoreLivesRemainingColor, TwoLivesRemainingColor, OneLifeRemainingColor are all set to their public constant defaults. Width and Height are set to their respective parameters. Limits are calculated (see CalculateLimits()).

Advanced Editor/Nine-Parameter constructor (char screengrassChar\_, char borderChar\_, ConsoleColor screengrassForeColor\_, ConsoleColor screengrassBackColor\_, ConsoleColor borderForeColor\_, ConsoleColor borderBackColor\_, int width\_, int height\_, int borderWidth\_) summary: ScreengrassChar, BorderChar, ScreengrassForeColor, ScreengrassBackColor, BorderForeColor, BorderBackColor, Width, Height, BorderWidth are all set to their respective parameter values. MoreLivesRemainingColor, TwoLivesRemainingColor, OneLifeRemainingColor are all set to their public constant defaults. Limits are calculated (see CalculateLimits()).

External Methods

Public void Render() summary: Takes no arguments. Draws the play area in the console based on its dimensions, characters and colors. Returns void.

Public void UpdateDisplay(int currentLevelIndex, Player player, Coin[] coins) summary: Takes the index of the current level in the levels array (see Program.cs.levels), an instance of the player object and an array of coins, and uses it to display the current round, number of coins collected, total number of coins, and the player’s lives remaining. Changes colors based on how many lives the Player instance’s backing field has (see LIVES\_REMAINING ConsoleColors). Returns void.

Internal Methods

Private void CalculateLimits() summary: Takes no arguments. Sets the UpLimit, DownLimit, LeftLimit, and RightLimit properties based on the dimensions of the Game Area itself, ensuring the player cannot clip outside of the game area itself. Returns void.