GcodeWriter.java

Overview

Among all the classes, the GcodeWriter class is the largest class and does most of the heavy lifting; it is a bridge between the GUI and the generation of Gcode commands, *.gcode* file, read by RepetierHost. Its main purpose is to construct the final printed solid by manipulating Polygon objects while performing materials change. This manipulation of Polygon objects takes place in the buildSolid( ) method aided by various helper methods like printFrame( ), cookFilledLayer( ), etc.

Please note that while most are used, not every variable of the GcodeWriter class is used to generate Gcode commands. Those unused variables are not included in this document but do have comments next to them in source code. Many of the variable values are taken in from the GUI through the initFromGUI( ) method.

Frequently used Gcode commands are as follows:

a. G01-type. Linear motion. Format: "*G01 X Y Z F E"* where X, Y and Z specify the 3D coordinates to move to, F specifies the travel speed, and E is extrusion coordinate on the filament.

b. M106- type. Turns on heating component. Format: *"M106 S"* where S specifies heat power (0-255 range).

c. G04-tyoe. Pausing. Format: "G04 P" where P specifies pause duration in milliseconds.

IO Variables	**Description**
private static String fileName	Name of output Gcode file, taken in from GUI
private static File file	File object
private static FileWriter outPut	FileWriter object that outputs strings of Gcode commands
private String filePath	Path to output Gcode file
Geometry Variables	**Description**
private static double spacing	Distance, in mm, between 2 nested frames
private static double twist_angle	Radian angle to rotate each layer by
private static double radius	Radius of layer polygon
private static double x_center	Center's x-coordinate of layer polygon
private static double y_center	Center's y-coordinate of layer polygon
private static int side_count	Polygon's number of sides
private static int top_thickness	Layer polygon's top layer thickness. Thickness is defined in terms of number of layers
private static int bottom_thickness	Layer polygon's bottom layer thickness. Thickness is defined in terms of number of layers
private static int bottom_layers	Number of bottom layers
private static double bed_z	Z-coordinate of the printing bed
final double PI = Math.PI;	Constant PI value
private static double layer_height	Height difference between 2 layer polygon

Extrusion-Control Variables	Description
private static double unit_E	Calculated using a formula inherited from past code. unit_E defines the coordinate unit on the extrusion rod.
private static double nozzle_dia	Constant of 1.8D inherited from past code
private static double extrusion_width = 1.5D * nozzle_dia;	Constant inherited from past code. Used in the calculation of unit_E
private double E	The current extrusion coordinate. The E value controls where on the extrusion rod to move to; It thereby controls how much material to extrude and how much distance to retract the plunge during material change.
Multimaterial Variables	**Description**
private static Material mat1	A material object representing the base material
private static Material mat2	A material object representing the secondary material, either powder or fillings
private static Material mat3	A material object representing some addition material, usually a second filling
private static Material currentMat	The material that is currently being printed with
private static double retraction	The amount to decrement the E value by during a retraction
private static double shakeSpeed	The speed at which the extrusion rod moves during a shake
private static int numOfShakes;	The number of up-down repetitions the extrusion rod performs during a shake
private static double shakeHeightOffSet;	The distance between the most recently printed layer and the bottom of the shaker syringe.
Cooking Variables	**Description**
private static double cook_y_offset	Y-coordinate offset distance of the heating component from the syringe.
private static double cook_temp	Heating component's power in the range of 0-255
private static double cook_lift;	The distance to move in the z-coordinate after cooking each layer. Current implementation doesn't take this into account, but maybe necessary for the future when material rises as it's being cooked
private static double cook_temp_standby	Heating component's power when not cooking (on standby)
private static double cook_frame_speed	The speed at which to move when trace cook
Speed-Control Variables	**Description**
private static double travel_speed	The speed at which to move when not printing
private static double print_speed	The speed when printing
private static double z_lift	The z-coordinate to move to after each layer print

File and Variables Setup Methods	Description
public void initFile(HashMap<String, String> settings)	Creates file-related objects and setups up named output Gcode file
public void initFromGUI(HashMap<String, String> settings, int option)	Initialize class variables with values taken in from GUI
public void closeFile()	Close files and writes Gcode command that brings tool head home
public String getFilePath()	Returns path to directory where Gcode file is output
Printing Helper Methods	**Description**
private void printFrame(Polygon p, double height)	At z-coordinate height, extrudes material as tool head traces out the perimeter of polygon p
private void fillLayer(Polygon polygon, double height)	At z-coordinate height, extrudes material as tool head traces out the nest polygonal frames of polygon. This has the same effect as filling in a layer from the perimeter inward.
Cooking Helper Methods	**Description**
private void cookFrame(Polygon p, double height)	At z-coordinate height, turns on heating component as tool head traces out the perimeter of polygon p
private void cookFilledLayer(Polygon polygon, double height)	At z-coordinate height, turns on heating component as tool head traces out the nest polygonal frames of polygon. This has the same effect as cooking a layer from the perimeter inward.
Multimaterial Support Methods	**Description**
private void pickUpMaterial(Material mat)	picks up material syringe *mat* from the material rack. The tool head may not be carrying a syringe when this method is called. pickUpMaterial() and dropMaterial() both use z_insert, which represents the Z-coordinate of the entry to the tool slot. This method retracts the extruder head, moves to Z-coordinate z_clear to not disturb the print, and aligns the extruder head with the material's slot 40mm away in the Y direction. It then lowers to the syringe's height, allowing the magnets to attach to the tool head, and raises back to z_insert, effectively lifting the syringe out of its slot. The tool head returns to z_clear, restores the extruder head to the current material's level in the syringe (E_curr), and updates the current material to be the one just picked up.
private void dropMaterial(Material mat)	Drops a material into its slot on the tool rack. It first updates the variable the stores the material's level in the syringe, retracts the extruder head, and clears in the Z dimension. It then aligns with the material's slot 40mm away, moves up to the slot, drops to the syringe's height, and moves backward 3mm. This motion is what locks the syringe into its slot, and the vertical bar keeps the syringe in place as the tool head moves backward as the magnets detach. The tool head then moves back to z_clear, and the current material is set to null because the tool head no longer carries a material.
private void spotDispensePowder(Material mat, double x, double y, double z)	Dispenses powder by shaking a syringe filled with powder. This method moves the extruder head to attach to magnets on the top of the shaker, moves the tool head to the desired coordinates, specified by x-y-z arugment, and moves the shaker's shaft up and down 3mm to complete one shake. The number of shakes is inputted by the user in the GUI.

Main Solid-Construction Method	Description

public void buildSolid(int option)	Takes in user's selected option in the GUI (from Single Material, Base-Powder, Triple Materials, Military Demo) to build a solid layer by layer. The function uses the option argument and calls printFrame(), fillLayer(), cookFrame(), cookFilledLayer(), pickUpMaterial(), dropMaterial(), and spotDispensePowder() as needed to build a complete solid with desired extra functionalities.
	
	Remarks on design decisions and local variables:
	a. double solidHeight = 2.0 x base Polygon's radius: solidHeight is chosen to be twice base radius for aesthetic reason. This is also why, in the GUI, the user-specified number of layers and base polygon radius should be the same. eg. default number of layer = 30 layers; default radius = 30 mm.
	b. double shrinkFactor is the ratio between the radius of a layer polygon and the radius of the one below it. To maintain the proportioned mentioned in a), it is calculated using the formula: shrinkFactor = (1.0 - ((solidHeight) / (2 * currentRadius * total_num_layers)))
	c. The outer most for loop controls the incrementation of layers. At the end of each iteration, the loops shrinks the current layer polygon and twist it by the angle taken in from the GUI before incrementing the height.
	d. The option argument has the following associated integers
	a. final int SINGLE = 1;
	b. final int DOUGHPOWDER = 2;
	c. final int CREAMCHEESE = 3;
	d. final int TRIPLEMAT = 4;