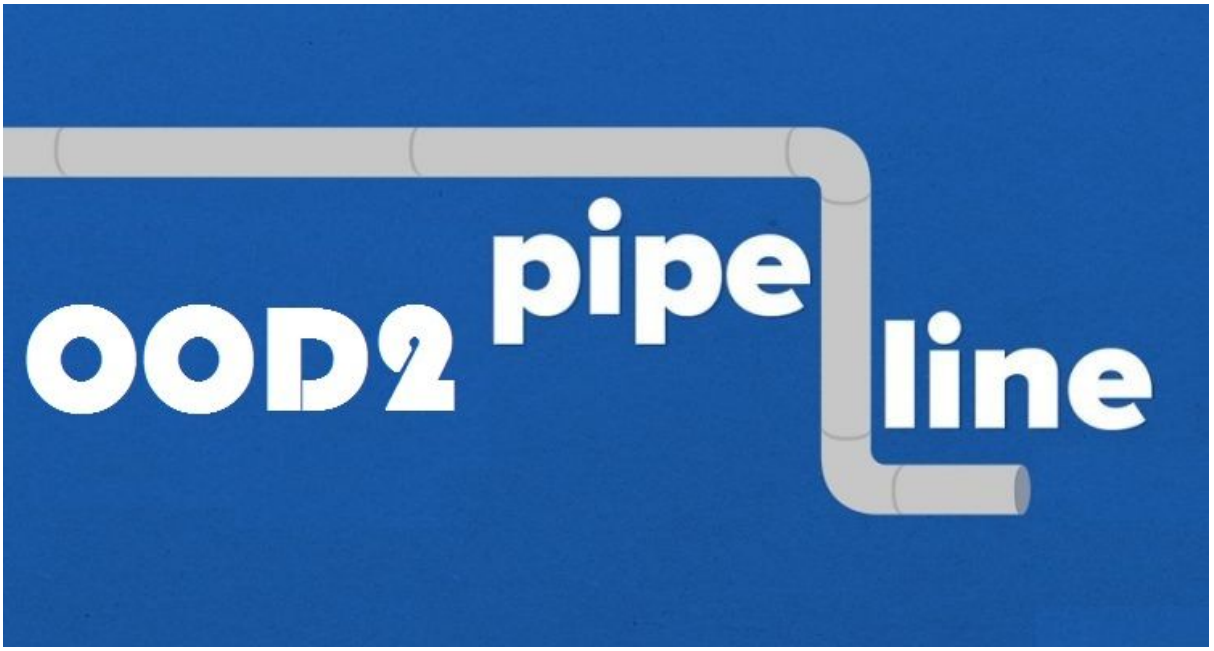


Design document

Date

12 - June - 2017.

OOD2 group

Team (names and student number)

Hoang Linh - 2233495

Fei Pei- 2585413

Simon Onumajuru - 2727897

Xiankuan Peng - 2624109

Status

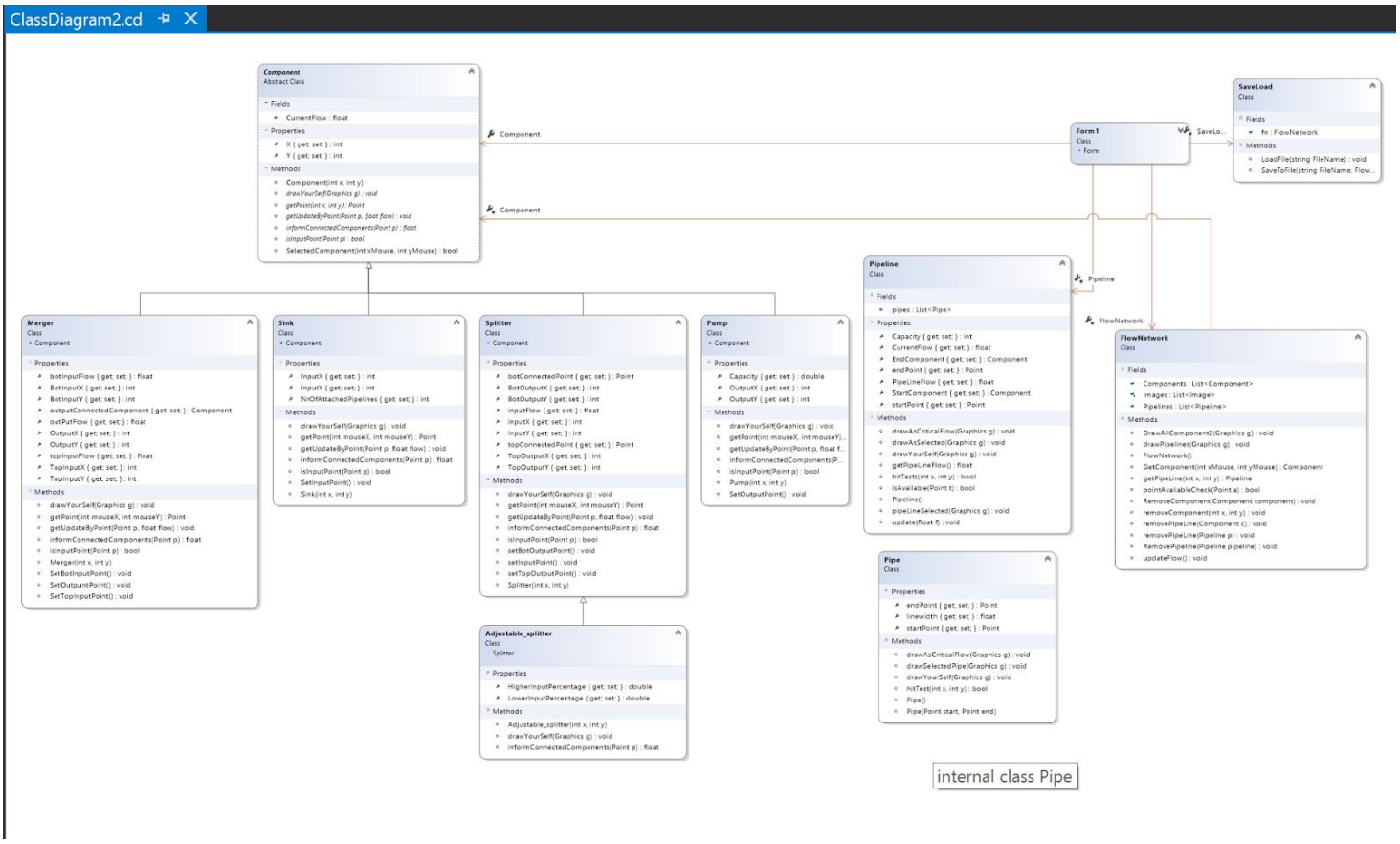
Final version draft.

INTRODUCTION

Introduction

This is a pipeline project to deliver an application where user can create his/ her own pipeline system and makes use of it. The application is called “Pipelines Simulator”, it provides possibilities for user to create and figure out the most efficient pipeline system to be used.

Class diagram:



COMPONENT CLASS

Constructors:

Name	Description
Component(x,y)	Initializing new instance of the component class

Field:

Name	Description
CurrentFlow	A field variable for the current flow

Properties:

Name	Description
X: int	X coordinates for the component
Y: int	Y coordinates for the component

Methods:

Name	Description
drawYourSelf(Graphich g): void	Drawable abstract method which allows each components to draw itself
SelectedComponent(x,y): bool	Returns true if user clicks on a drawn component(This is used in the method getComponent() in the FlowNetwork to select the component.
getPoint(x,y) : Point	Abstract method, return a point based on x,y and subclass.
isInputPoint(Point p): bool	Abstract method, return true if point p is on the left side of the component
informConnectedComponents(Point p): float	//dead end, not implemented
getUpdateByPoint(Point p, Float flow): Void	Abstract method,inputFlow will be assigned with the value of float flow depends on the position of point p

MERGER CLASS

Constructors:

Name	Description
Merger()	Initializing new instance of the merger class

Properties:

Name	Description
TopInputX	X coordinate of the Top layer of merger.
TopInputY	Y coordinate of the Top layer of merger.
OutputX	X coordinate of Output pipeline for merger
OutputY	Y coordinate of Output pipeline for merger
topInputFlow	Stores the top input flow for the merger
botInputFlow	Stores the bottom input flow for the merger
outPutFlow	Stores the output flow for the merger
BotInputX	X coordinate of the Bottom layer of merger.
BotInputY	Y coordinate of the Bottom layer of merger.

Methods:

Name	Description
SetTopInputPoint(): void	Connects and sets the top input point
SetBotInputPoint(): void	Connects and sets the bottom input point
SetOutputPoint(): void	Sets output points to connect
getPoint(x,y)	retrieves the point position clicked on the canvas , return topInput point,botInputPoint or outPutPoint depends on x and y.
isInputPoint(Point p)	Return true if point p is in the left of the component
informConnectedComponents(Point p)	//dead end, not implemented
getUpdateByPoint(Point p, Float flow)	Assigned the value of flow to topInputFlow or botInputFlow depends on point p ‘s position
drawYourSelf(Graphich g): void	Allowing the class to draw itself

SINK CLASS

Constructors:

Name	Description
Sink()	Initializing new instance of the sink class

Properties:

Name	Description
InputX	X coordinate of the connected component
InputY	Y coordinate of the connected component
NrOfAttachedPipelines	The number of pipelines connected

Methods:

Name	Description
drawYourself(Graphich g): void	Allowing the class to draw itself
getUpdateByPoint(Point p, Float flow)	Always assign the current flow with the value of float flow
SetInputPoint(): void	Sets input points to draw pipeline
getPoint(x,y)	retrieves the point position clicked on the canvas ,always return a point with coordinate(inputX,inputY)
isInputPoint(Point p)	Always return true
informConnectedComponents(Point p)	//dead end, not implemented

SPLITTER CLASS

Constructors:

Name	Description
Splitter()	Initializing new instance of the splitter class

Properties:

Name	Description

TopOutputX	X coordinate of the Top layer of merger.
TopOutputY	Y coordinate of the Top layer of merger.
InputX	Splitter's input pipeline X coordinates
InputY	Splitter's input pipeline Y coordinates
InputFlow	Y coordinate of the Bottom layer of merger.
BotOutputtX	X coordinate of the Bottom layer of merger.
BotOutputtY	Y coordinate of the Bottom layer of merger.
topConnectedPoint	Top connected point of the merger
botConnectedPoint	Bottom connected point of the merger

Methods:

Name	Description
SetInputPoint()	Sets the input point to draw pipeline
SetOutputPoint()	Sets the output point to draw pipeline
setBotOutputPoint()	Sets both output points to draw pipeline
GetFlow(p)	Gets the current flow of the splitter
drawYourself(Graphich g)	Allowing the component to draw itself
getPoint(x,y)	retrieves the point position clicked on the canvas ,return a point with coordinate of (inputX,inputY) or (botOutPutX,botOutPutY) or (topOutPutX,topOutPutY) depends on x,y
isInputPoint(Point p)	Return true if p is in the left of the component
informConnectedComponents(Point p)	//dead end, not implemented
getUpdateByPoint(Point p, Float flow)	Assigned the value of float flow to inputFlow

ADJUSTABLE SPLITTER CLASS

Constructors:

Name	Description
AdjustableSplitter()	Initializing new instance of the adjustable splitter class

Properties:

Name	Description
UpperFlow	Determines the upper flow of the splitter
HigherInputPercentage	Percentage input for the top point
LowerInputPercentage	Percentage input for the bottom point

Methods:

Name	Description
drawYourself(Graphich g): void	Allowing the class to draw itself
informConnectedComponents(Point p)	Informs the connected components of the bottom input flow and top input flow

PUMP CLASS

Constructors:

Name	Description
Pump()	Initializing new instance of the pump class

Properties:

Name	Description
OutputX	X coordinate of the component connected to output port
OutputY	Y coordinate of the component connected to output port
Capacity	Capacity of the flow

Methods:

Name	Description
DrawYourself(Graphich g): void	Allowing the class to draw itself
SetOutputPoint()	Sets the output point to draw the pipe
getPoint(x,y)	retrieves the point position clicked on the canvas
isInputPoint(Point p)	Return false
informConnectedComponents(Point p)	//dead end, not implemented
getUpdateByPoint(Point p, Float flow)	throw new NotImplementedException();

SAVELOAD CLASS

Methods:

Name	Description
LoadFile(Filename:string): void	Loading the project, This method loads a network by reading text file and creating new items of the network line by line.
SaveToFile(Filename:string, f: FlowNetwork): void	Saving the project, This method saves the entire network (which is found on the canvas- connected or not) into a file by writing down each element

FLOWNETWORK CLASS

Constructors:

Name	Description
FlowNetwork()	Initializing new instance of the FlowNetwork class

Properties:

Name	Description
Component: List	List of Components
Images: Images	List of Images loading all the components of the flow network
PipeLine: List	List of PipeLines

Methods:

Name	Description
getComponent(x,y): Component	returns a component that user clicked on, otherwise show warning to the user.
getPipeLine(x,y): PipeLine	returns a pipeline that user clicked on (by calculating the points which is a coordinate x and y to obtain the value)
DrawAllComponent2(Graphics g)	Each components draws itself separately
removeComponent(Component c): void	Removes a component
removePipeline(Pipeline p): void	//not implemented,
drawPipeLines(Graphic g): void	Loop through list pipelines, and call drawYourself on each

	pipeline
RemoveComponent(x,y)	Removes all associated pipelines of the component with coordinate(x,y), and then remove that component from the list.
pointAvaliableCheck(Point a)	Detects if there is an available point to connect , attached to components
removePipeline(Component c): void	Removes a pipeline associated to a component
updateFlow()	Updates the flow of the pipelines

PIPELINE CLASS

Constructors:

Name	Description
PipeLine()	Initializing new instance of the PipeLine class

Properties:

Name	Description
isSelected	
startPoint	Y coordinate of the first connected component
endPoint	X coordinate of the second connected component
Pipes: List<Pipe>	List of pipes
StartComponent: Component	First connected component
EndComponent: Component	Second connected component
CurrentFlow	has the current flow of the pipeline

Methods:

Name	Description
drawYourSelf(Graphics g)	Allowing the class to draw itself
drawAsSelected(Graphics g)	Foreach pipe in pipes, calling pipe's drawAsSelected method
pipeLineSelected(Graphics g)	Draw all pipes in list pipes in black
isAvailable(Point t)	this will loop through all pipe in pipes, return true if point t is not used

	as a startpoint or an end point
hitTests(int x, int y)	call hitTest on each pipe in the pipes list,return false if any hitTest from pipe return false
getPipeLineFlow()	return the output flow from the source component.
update(float f)	update the end component's current with float f

FORM CLASS

Method:

Name	Description
ExportToPNG	Exports the draw flow network into a PNG (picture) format

PIPE CLASS (Extra Class)

Constructors:

Name	Description
Pipe(Point start, Point end)	Initializing new instance of the pump class

Properties:

Name	Description
startPoint	Store start point
endPoint	Store end point
linewidth	Indicates the detection distance between a point and the pipe(its a line)

Methods:

Name	Description
drawYourSelf(Graphics g)	Allowing the class to draw itself
drawAsSelectedPipe(Graphics g)	Allowing the class to draw the pipe

hitTest(x,y)	Calculate the distance between the point(x,y) and the pipe, return true if its smaller than linewidth/2

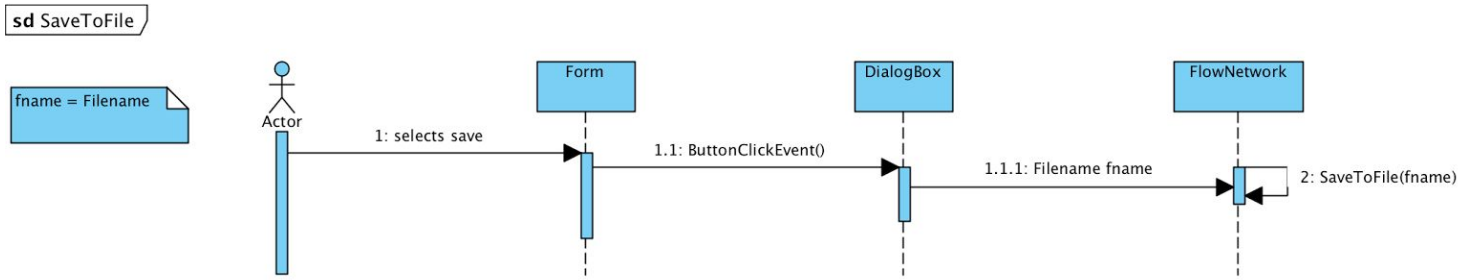
FORM PROPERTIES (Extra class)

Method:

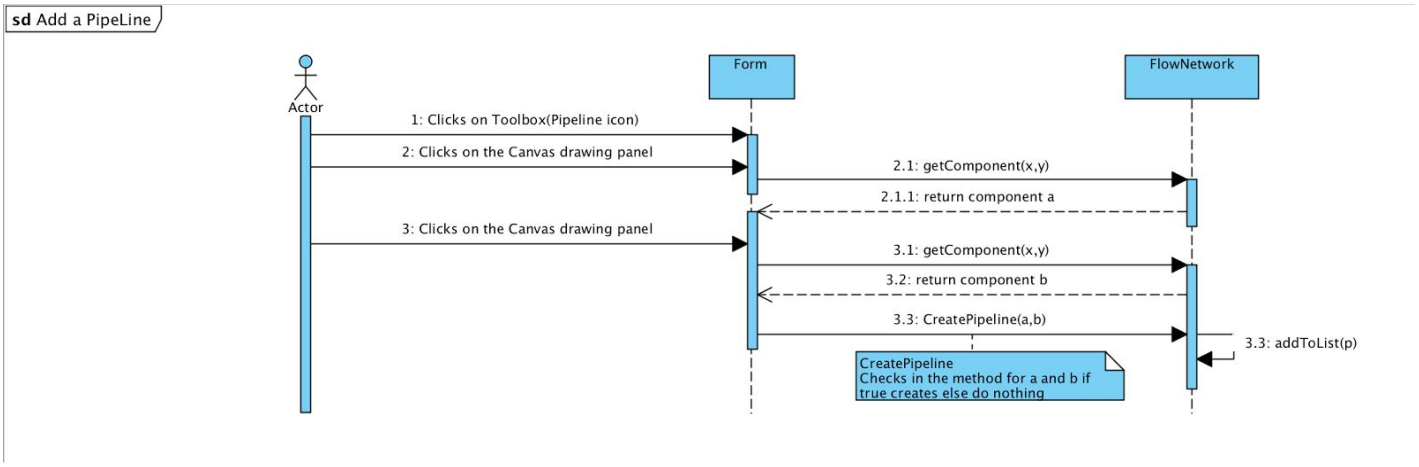
Name	Description
selectedComponent	Component that user currently selected
selectedPipeline	Pipeline that user currently selected
tempPoint	Store the latest point when creating a pipe
tempPipe	An empty/unfinished pipe.
tempPipeline	An empty/unfinished pipeline

Sequence diagram:

SaveToFile:



Add PipeLine:



Remove Component: