點構式向量繪圖系統

系所／資訊工程學系

指導老師／無

組員／周孝倫

現今的向量繪圖，大多是基於函數的形狀所構成，較不強調形狀與點之間的關係，本專案旨在嘗試一種全新的繪圖邏輯，以點為核心，函數為輔助構成物件的形狀。

本繪圖邏輯系統將點作為一個形狀的根基，其中點分為動態點（dynamic point）與靜態點（static point），動態點在繪圖邏輯中只被引用，不被改變，可以藉由動態點的位置作為函數的變數輸出形狀，是經典形狀的基本構成元素，如多邊形、貝茲曲線、貝茲曲線構成面、折線等等；靜態點則在繪圖邏輯中被設置參數，可作為整體形狀的外框，可用於做出始化的凸包運算、相機聚焦參考系、中心位置標記錨點等。

本繪圖邏輯以基於Java Swing框架，設計之繪圖平台實現，名為「Painter」，其中以Scene類別作為整體系統的平台，提供一般應用模式以及檢視模式，兩種開啟的入口。

「ExportLoadSystem」則提供為此架構設計獨立的檔案協定格式，並支援檔案的輸入及自訂附檔名「.vecf」輸出，如圖1。

本程式具備多作業系統的運作，支援Linux Ubuntu以及Windows，以及具備Java JDK-25環境之平台使用，並使用Java撰寫程式，確保每個平台同一版本之間毫無任何差異。

在平台的UI/UX設計中，引用了各式快捷鍵的設計、直覺操作的圖層管理與工具列，提高使用者上

手的速度，並具備詳細的 readme檔案以及清晰的物件導向原則與明確的設計結構以及必要的API接口，如圖2。

對於形狀的設計擁有高度的彈性，降低後續功能擴張的成本。除了繪圖邏輯與點的設計，也能覆寫放大縮小、中心點定義、移動操作等等動作。

本架構之設計以「PainterObj」類別為物件之核心，其中可延伸出各種物件如群組、基本圖形、PNG圖檔等等，並接受動態點與靜態點參與形狀形成，忽略邊界概念，並引入實體描述規則API定義可選取的區域，此處可引用靜態點之概念，符合本專案構思之繪圖邏輯架構之精神。

本圖應用程式介面具備托拽控制之圖層管理，上列工具欄之快捷圖案生成以及多邊形、折線、貝茲曲線、貝茲曲線構成面之圖形，藉由動態點與靜態點集合之凸包實時呈現於圖層管理器，使圖層管理器之使用，並且具備群組與解散群組功能，實現集體操作，如圖3。

一張含有 文字, 螢幕擷取畫面, 作業系統, 軟體 的圖片

AI 產生的內容可能不正確。

圖1：檢視器開啟之畫面

預覽器可以使用兩種方式開啟：直接連續點擊兩下開啟並拖拽檔案入畫面，以及直接透過檔案開啟，使用者在開啟.vecf檔案時可預設本程式之檢視器開啟。

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

AI 產生的內容可能不正確。

圖2：應用程式使用者介面

本介面之上方工具列可透過滑鼠托拽，改變工具欄之高度，並具備初始圖案；三角形、矩形、圓形之一鍵生成以及多點形狀生成，以及群組及解散群組之功能；右方圖層管理器亦可滑鼠拖拽控制寬度；其餘操作有右鍵點擊針對選取物件上色、Ctrl與C同時按下複製所選取之物品、Ctrl愈A同時按下選取全部物品、按下Ctrl並選取多個物件或點進行同步操作，並可藉此拖動物件的邊、Ctrl與Z或Y鍵可復原或重做，本程式最多紀錄100筆操作資料。Shift鍵按下進行水平或垂直精準拖動，上下左右方向鍵控制物件大小、滾輪控制化面縮放、Ctrl與S鍵按下存檔。

一張含有 文字, 螢幕擷取畫面 的圖片

AI 產生的內容可能不正確。

圖3：軟體設計之架構概念圖

本程式基於點構式向量繪圖邏輯，設計以Scene為核心、以「PainterObj」為物件、

「ExportLoadSystem」為檔案管理器、其餘為繪圖用之工具之架構原則，確保後續所有接手設計都能以較低成本改良設計。