

Lab 6

basic network programming

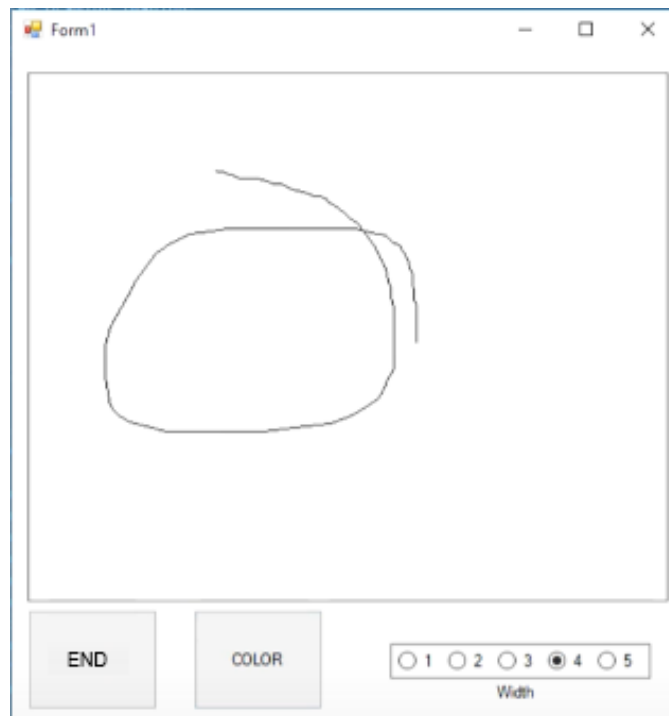
Bài tập tổng hợp

A. NỘI DUNG THỰC HÀNH

Sinh viên chỉ cần chọn một trong các ứng dụng sau để thực hiện. Cho phép thực hiện hình thức cá nhân hoặc nhóm 2 thành viên.

1. Whiteboard

Whiteboard (bảng trắng) là một công cụ khá cần thiết trong quá trình làm việc nhóm giúp dễ dàng chia sẻ ý tưởng đến những thành viên khác. Bạn hãy xây dựng một ứng dụng cho phép nhiều người dùng có thể cùng xem nội dung trên bảng và thực hiện các thao tác viết, vẽ, xoá.



Yêu cầu:

- Xây dựng ứng dụng dựa trên mô hình Server – Clients
- Bất kì client nào cũng có thể sử dụng chuột để vẽ theo ý muốn. Khi có một thay đổi nào trên Whiteboard, tất cả clients sẽ tiến hành cập nhật sự thay đổi này (realtime).
- Khi một client bất kỳ nhấn vào nút End (kết thúc), tại mỗi client sẽ lưu lại hình ảnh hiện tại trên Whiteboard thành file .PNG hoặc .JPG và tự động đóng chương trình.
- Cho phép client tùy biến màu sắc và độ đậm của nét vẽ.
- Hiển thị thông tin số lượng client đang kết nối.

Yêu cầu mở rộng:

- Cho phép chèn hình ảnh từ Internet vào whiteboard (khuyến khích phát triển tính năng có thể thay đổi kích thước ảnh khi chèn vào).
- Khi client tham gia, cập nhật whiteboard trên form của client như whiteboard hiện tại.
- Khi số lượng client đạt đến giới hạn (vd 5), gửi email cảnh báo đến quản trị viên.

Gợi ý:

- Sinh viên có thể tham khảo ví dụ sử dụng Drawing trong C# tại <https://www.youtube.com/watch?v=SFwBGVcxaRo>

2. GAME ĐOÁN SỐ

Thực hiện viết trò chơi đoán số đơn giản cho phép nhiều người cùng tham gia. Server sẽ chọn một số nguyên ngẫu nhiên x (giữ bí mật) trong phạm vi ngẫu nhiên $a < x < b$ và thông báo cho người chơi phạm vi số cần tìm. Người chơi sẽ đưa ra các con số dự đoán của mình. Người chơi đoán đúng và nhanh nhất là người chiến thắng, các đáp án của người chơi khác sẽ bị bỏ qua và bắt đầu một lượt chơi mới.

Yêu cầu:

- Xây dựng ứng dụng trên mô hình Server – Clients.
- Bên phía Clients, thiết kế giao diện cho phép điền tên người chơi, điền con số dự đoán và thông báo của server (lượt chơi mới bắt đầu, người thắng cuộc, phạm vi số cần đoán,...)
- Bên phía Server thiết kế giao diện hiển thị các thông tin số đang cần tìm, phạm vi số đang cần tìm, lịch sử trò chơi, số người đang tham gia, số lượt chơi.
- Giới hạn khoảng thời gian giữa 2 lần gửi kết quả dự đoán tối thiểu 3 giây và hiển thị thời gian đếm ngược trên Form của client.
- Phát triển tính năng tự động chơi, cho phép client sinh ngẫu nhiên kết quả và tự động gửi (các kết quả tự sinh ra không trùng với kết quả đã thử trước đó và thuộc phạm vi con số cần tìm).
- Server tự quyết định số lần chơi (tối thiểu là 5) để kết thúc trò chơi. Khi trò chơi kết thúc, phía clients sẽ lưu toàn bộ lịch sử thông báo của server thành file history.txt, server gửi toàn bộ lịch sử trò chơi lên website <https://ctxt.io> (hoặc các trang web khác có chức năng tương tự); sau đó cả clients và server tự động đóng ứng dụng.

Yêu cầu mở rộng

- Khi kết thúc trò chơi, thông báo người thắng chung cuộc (dựa trên các tiêu chí: có số lượt trả lời đúng nhiều nhất → có tổng số lần dự đoán sai ít nhất).
- Thêm tính năng tính điểm cho người chơi (ví dụ: trừ điểm mỗi lần đoán sai, cộng điểm mỗi lần đoán đúng)

B. YÊU CẦU & ĐÁNH GIÁ

1. Yêu cầu

- Sinh viên thực hành và nộp bài theo **cá nhân hoặc nhóm 2 người** tại website môn học theo thời gian quy định
- Bài nộp bao gồm toàn bộ Source-code của ứng dụng, clip demo hoạt động của ứng dụng, hướng dẫn các bước để thực thi chương trình.

Toàn bộ project đặt vào 1 file nén (.zip) với tên theo quy tắc sau:

MSSV-HọTênSV

Ví dụ: 16520067-NguyenVanA

2. Đánh giá kết quả

- Tiêu chí đánh giá:
 - Chương trình chạy được, hoàn thành các yêu cầu cơ bản: **+70%**.
 - Có kiểm tra các điều kiện ràng buộc khi nhập dữ liệu, code "sạch" [2], đặt tên biến rõ ràng; báo cáo trình bày rõ ràng: **+30%**.
 - Nộp bài không đầy đủ; lỗi, không chạy được; nộp trễ; sao chép code nhóm khác, nguồn có sẵn: **xử lý tùy theo mức độ (-10 – 100%)**

C. THAM KHẢO

- [1] Microsoft (2018). C# Guide. [Online] Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/>
- [2] Martin, R. C. (2009). ***Clean code: a handbook of agile software craftsmanship***. Pearson Education.
- [3] Network Programming in the .NET Framework: <https://docs.microsoft.com/en-us/dotnet/framework/network-programming/>

HẾT