CI/CD Strategy and Testing

CI/CD Testing Strategy:

Our project incorporates Continuous Integration (CI) and Continuous Deployment (CD) using GitHub Actions.
This strategy ensures that every change to the codebase is tested, built, and deployed in an automated and reliable manner.

The CI/CD approach minimizes manual errors, shortens feedback loops, and allows for faster and safer delivery of new features or fixes.

Why CI/CD is Important:

- Ensures code quality through automated testing before merging.
- Provides early feedback to developers.
- Enables rapid deployment to multiple platforms.
- Reduces risk of bugs in production.
- Saves time and increases team efficiency.

CI Workflow (Build and Test):

The CI workflow automatically triggers on each push or pull request to the main branch. It installs necessary dependencies like Qt and X11, builds the project using qmake and make, and runs the application and tests in a headless virtual display environment (using xvfb). This ensures that the project compiles and runs properly after every code change.

Key steps in CI workflow:

- - Checkout the source code.
- - Install Qt and other dependencies.
- - Build the project using qmake and make.
- - Run the application in a headless environment.
- - Execute automated tests to verify functionality.

CD Workflow (Release Process):

The release workflow is triggered when a version tag (e.g., v1.0.0) is pushed or manually through the GitHub UI.
This process builds the application for three major platforms: Windows, macOS, and Linux using a matrix strategy.
It then packages the output into platform-specific formats such as AppImage (.AppImage), Windows executable (.exe), or macOS image (.dmg), and uploads them to the GitHub Releases page.

Key steps in CD workflow:

- - Detect version tag or manual release trigger.
- - Install platform-specific dependencies.
- - Build the project using CMake and Qt.
- - Create installer packages for each OS.
- - Upload the packaged binaries to GitHub Releases.

Summary:

With CI/CD integration, our Advanced Tic Tac Toe project maintains high reliability, repeatability, and cross-platform support throughout development and delivery. This pipeline not only automates testing and packaging, but also provides a professional-grade deployment mechanism for our users.