Validate only changed katas…

Table of Contents

Background

Currently, the notebook validation step takes about 30 minutes. We validate all katas on every CI run.

This is really not required for katas which are not changed.
Eg: A person making changes in BasicGates kata can never break any other kata like GraphColouring. But sometimes CI fails on katas where no changes have been made because some tests are random and have a passing probability of approx 99%.

After this change, only katas that have been changed will be validated by CI. This has mainly 2 benefits :

- Shorter CI times
- Improved CI reliability

Approach

- Merge microsoft/QuantumKatas/main branch to the current branch for which the pipeline is being run. In case of merge conflict, pipeline run will fail.
- Get a list of changed files by running **"git diff"** and find the folder whose files changed. This helps us to get the list of katas for which we need to run a test.

  Kindly, refer to Current scope for validation of changes in below folders and Current scope by changes according to file name/type to know more about the current scope

- Validate only the specific katas that people have changed.

Execution plan

The plan is to implement a basic version of this in 3 PRs :

First PR - Validate katas instead of notebooks

- Firstly, modify "validate-notebooks.ps1" script to accept the list of katas (folder) instead of a single notebook. This script can be renamed to "validate-katas.ps1" to reflect this new intent. For each kata(folder) verify all the .ipynb files belonging to the kata(folder and subfolder(s)).
- List of katas would be supplied by "list-katas.ps1". After this PR, we would still be validating all notebooks just as we do now.

Second PR(Good to have) - Dynamic partitioning of katas

- Change existing sharing behavior based on threshold.

- Currently, we use start and end Index to control the number that are executed in each partition. This would make it a lot difficult to maintain after this change.

- Maybe we can modify the existing logic to share katas equally in N partitions dynamically at runtime. So, "list-notebooks.ps1 " will accept only one parameter(numberOfPartitions=2 currently).

Third PR - Validate only changed katas

- Change "list-katas.ps1" to only send the list of changed katas.

Open questions

Question 1: csproj file change

- If **csproj** files change, should we validate those katas? In many cases, we are concerned with if katas are compiling successfully.
- In cases of breaking changes, like this PR(https://github.com/microsoft/QuantumKatas/pull/468), which added <IQSharpLoadAutomatically> attribute in all katas; running a subset/ all katas might help.

Question 2: Dynamic threshold...

Do we want to change existing partitioning behavior based on threshold? This is in with reference to Second PR(Good to have) : Dynamic partitioning of katas in execution plan.

Advantages

- This partition logic would be generic enough that we would be able to change the number of partitions in case the number of katas grow significantly in future.
- No need to reset the threshold for better pipeline utilization with the addition of new katas.

Disadvantages

- Maybe this is a bit of over engineering :-). If we won't do this change then only the first partition would execute katas. In current threshold based behavior, only the first partition would execute the necessary katas.

Appendix

Current scope for validation of changes in below folders

- **Quickref/** -> no validation required
- **Scripts/** -> Run all kata validation only if “validate-notebooks.ps1” is changed. Can be updated in future to run only a subset of notebooks.
- **QuantumKatas root folder** -> No validation required. Important edge case to handle, otherwise all notebooks will run.
- **utilities/** -> We can skip validating katas if any file is changed here. Changes in these folders are either covered by Unit tests or validation requires some extra pipeline changes.

  Contains 5 sub-folders :
  - **Common/** and **CounterSimulatorTests/**–> “Common/” contains 3 files namely : “CounterSimulator.cs”, “Utils.qs”, “Delay.cs”
  - **DumpUnitary/ and DumpUnitaryTest/**
  - **Microsoft.Quantum.Katas/** - Contains KataMagic and CheckKataMagic code. No need to run separate validation of notebooks. Issue https://github.com/microsoft/QuantumKatas/issues/742 is tracking the validation for changes done to Magics

Current scope by changes according to file name/type

- ***.cs, *.qs, *.csproj** -> Validate the kata for which these files changed
- **Scripts/validate-notebooks.ps1** -> This impacts the test execution. We can all for now. Can be updated in future to run only a subset of katas. Here, is a PR(https://github.com/microsoft/QuantumKatas/pull/515) that aimed to improve perf by changing this file.
- Scripts/list-notebooks.ps1 -> No validation of katas

Possible Future improvements

- **Iqsharp-config.json, README.md -** no kata validation required
- **Reference_Implementation.qs** -> Run only the front-end while skipping the workbook.
- **Tasks.qs** - Check if project is compiling
- **Tests.qs** - Run both front-end and workbook
- **Changes in markdown cells in jupyter notebook** - No validation required.