

# Coding checklist

## Version control

1. use a version control system to keep track of changes, push to a remote for backup
2. commit often
3. write easy to understand commit messages (what changed and why it changed)
4. remove all `Console.WriteLine()` or `Console.out.WriteLine()` calls before committing
5. remove all commented out code before committing
6. have code refactoring turned on on file-save or before-commit

## Naming and spacing

7. use spaces for indentation
8. avoid lines longer than 120 characters
9. use English for all names in code
10. never use one letter names in code (outside for loop counters)
11. start boolean variables with is, should, do to be quickly understood (e.g. `isPlaying` , `shouldCleanup` )
12. name variables after what information they are holding
13. name functions after what they will be doing
14. avoid spelling mistakes
15. avoid abbreviations in names as they are not universally understood (acronyms are usually ok though, as they are better defined and more easily searched for)
16. write self-documenting code, only comment on weird, not-obvious statements

## C# code style

17. adhere to the C# general coding style (e.g. braces on newlines, `PascalCase` for methods)
18. always have braces, even around one line bodies
19. have one class per file
20. have classes in a logical directory structure
21. refactor long chunks of code into separate methods
22. use empty newlines to break up a blob of code into logical chunks
23. use one statement per line
24. do not import static methods, import their class and use them from it (e.g. `Class.StaticMethod()` )
25. extract magic constants into actual class constant ( `readonly` in C#) members
26. encapsulate well, only use public where necessary (unfortunately *Avalonia* requires a lot of public properties)
27. use early-return and early-break where applicable
28. try to compile without warnings, leave an explanatory comment if the warning is wrong and needs to be disabled
29. use `var` wherever possible and save time by letting the compiler determine the type
30. do not forget `Equals` and `HashCode` method if a class is used inside a List or a Set