PseudoCode Lab, VS Setup, Exercise 1 Q&A

Software Engineering 1

Dr Swen E. Gaudl

University of Plymouth 2022

COMP10000

Mobile.Plymouth.ac.uk

Please Sign in using Code: _ _

@https://durofy.com/mind-map-inheritance

COMP1000 Agenda This Week:

- VS Setup
- Pseudocode detailed look
- Q & A Exercise1

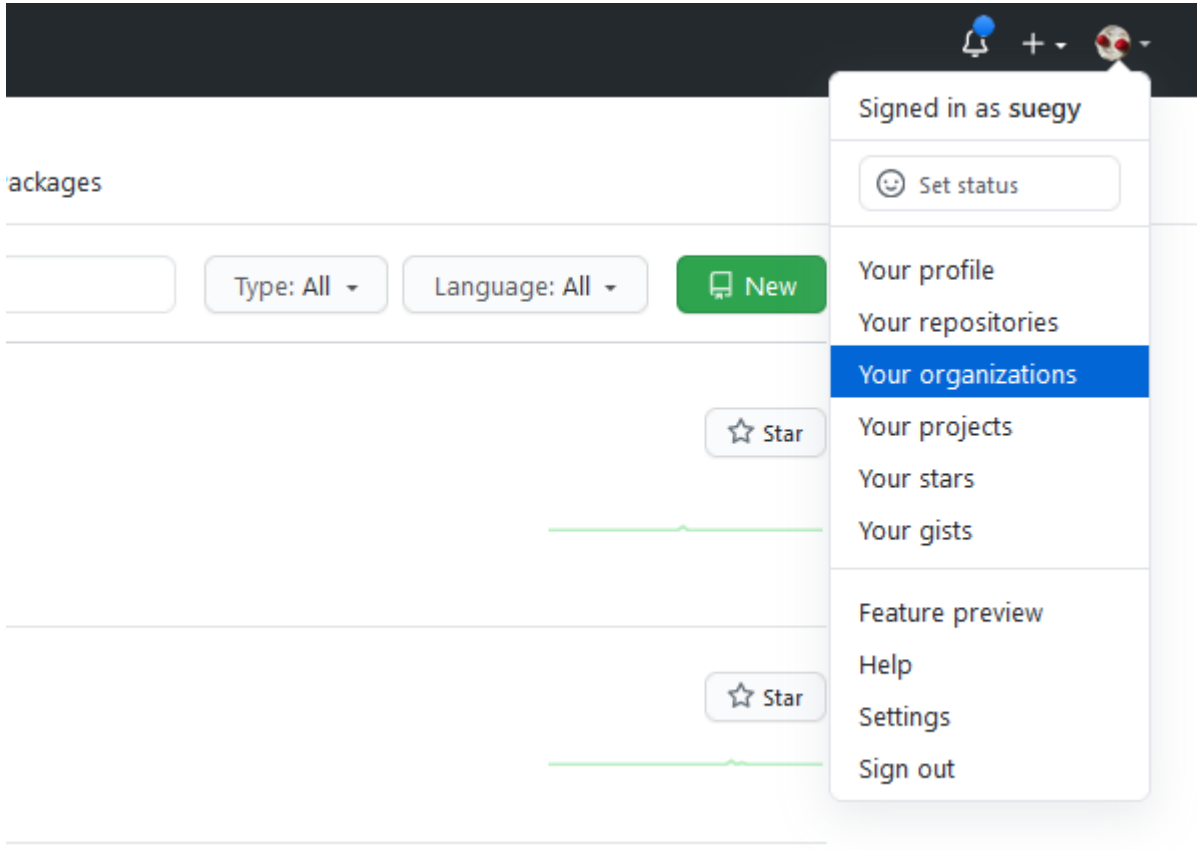
Recap and Intro to Version Control

Recap and Intro to Version Control

- What is version control?
  - What is a git?
- How should I use it?
- What makes my life easier?

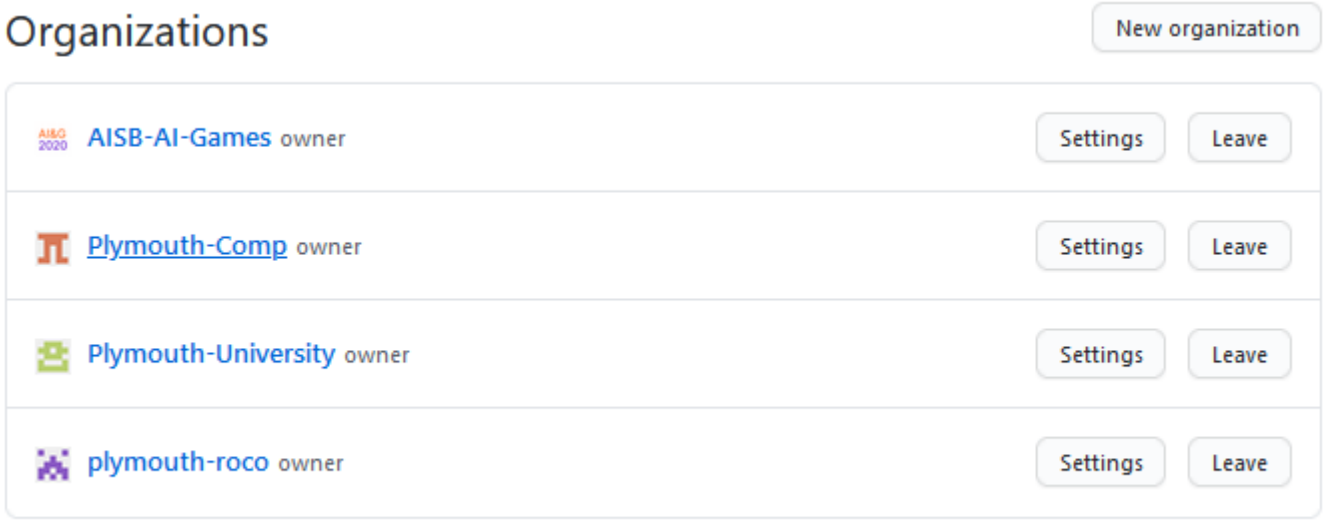
Introduction to Git

- How to start a project:
  - On github create a repository or for the classroom accept an invitation
  - Sign into your github account
  - Go to your organizations:



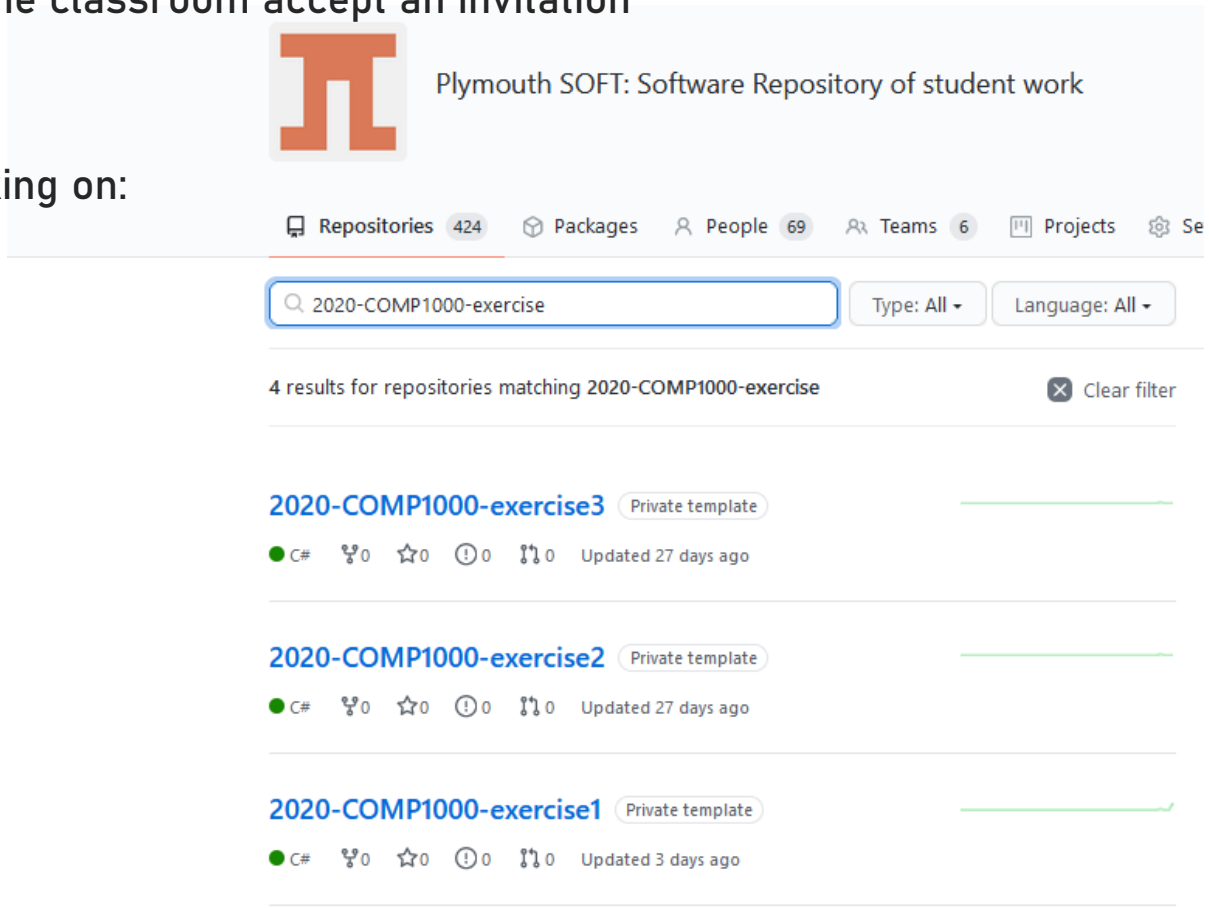
Introduction to Git

- How to start a project:
  - On github create a repository or for the classroom accept an invitation
  - Sign into your github account
  - Go to your organizations:
  - Go to Plymouth University:



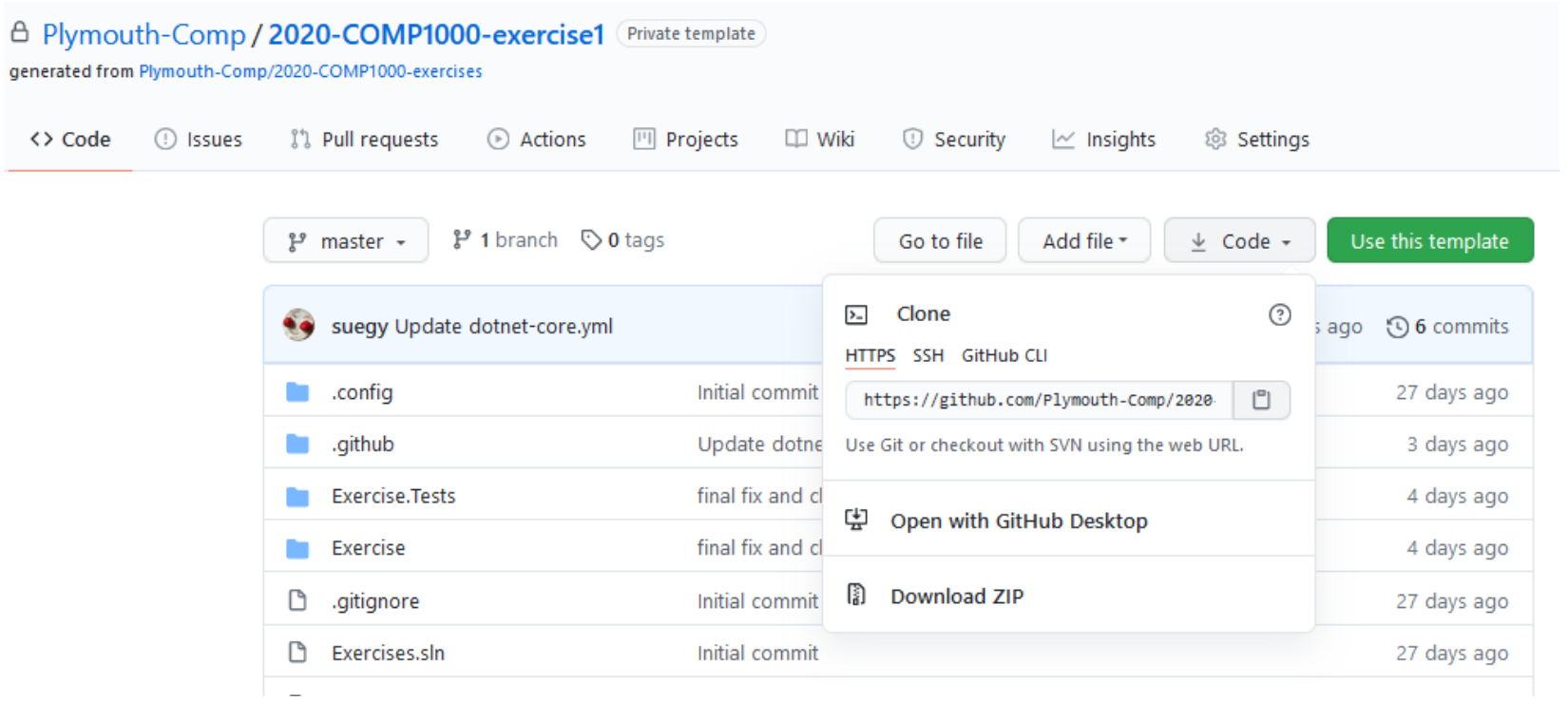
Introduction to Git

- How to start a project:
  - On github create a repository or for the classroom accept an invitation
  - Sign into your github account
  - Go to your organizations:
  - Pick/Search the project you are working on:



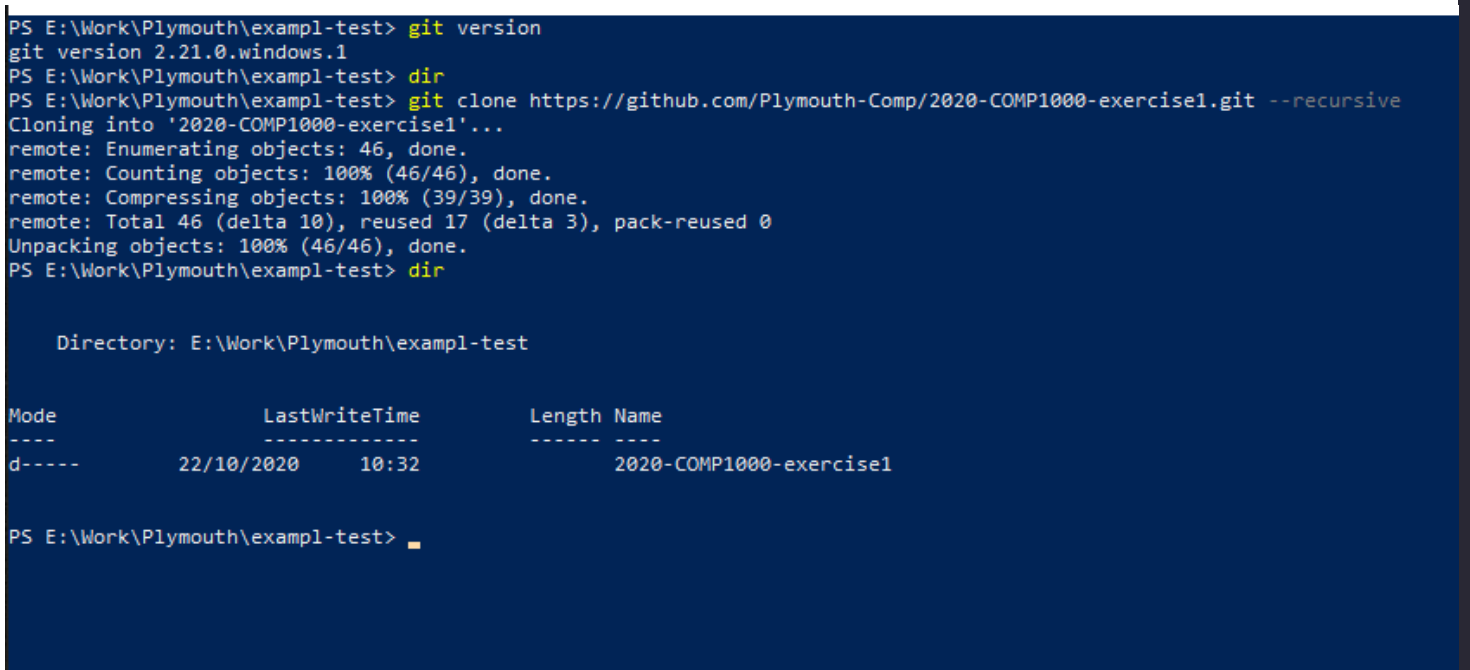
Introduction to Git

- How to start a project:
  - On github create a repository or for the classroom accept an invitation
  - Sign into your github account
  - Go to your organizations:
  - Pick/Search the project you are working on:
  - Go to Code:
    - Select https
    - Copy the URL



Introduction to Git

- How to start a project:
  - On github create a repository or for the classroom accept an invitation
  - Sign into your github account
  - Go to your organizations:
  - Pick/Search the project you are working on:
  - Go to Code
  - Open a Shell
  - Check git:
    - *git version*
  - *git clone <url> --recursive*
  - *CD* into the folder
  - Use *dir* to inspect folder



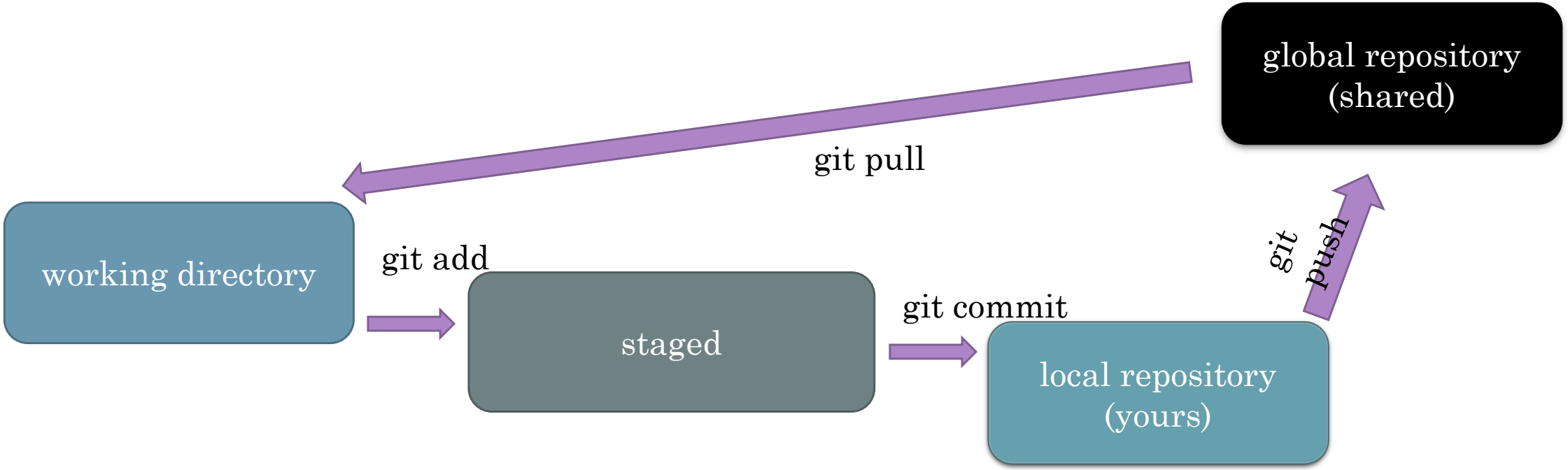
Introduction to Git

- How to start a project:
  - On github create a repository or for the classroom accept an invitation
  - Sign into your github account

- Working Routine (each day/change):
  1. Pull (git pull)
  2. work
  3. stage selected files (git status; git add <changes>)
  4. commit changes: (git commit –m “message”)
     1. write meaningful change message
     2. first line give a subject than add empty line before describing things in detail
     3. treat it like a diary
     4. if a milestone was reached: git tag
  5. back to 2
  6. before ending a day push (git push)

Personal Access Token and Git

- Why do I need to do that?
  - Howto setup and Work With: link



Visual Studio

preparing for the seminars/ programming:

1. on Github:
   - create an account
   - Accept invitation for COMP1000 exercise1 on the DLE
   - Deadline: 24/10/2022 10am GMT

Git Workflow

- Important things to remember
  - folders are difficult to work with
  - file renaming and deleting/moving problematic
  - use of external libraries in your project (large amounts of needed which does not change)

- Working Routine (each day/change):
  1. Pull (git pull)
  2. work
  3. stage selected files (git status; git add <changes>)
  4. commit changes: (git commit –m “message”)
     1. write meaningful change message
     2. first line give a subject than add empty line before describing things in detail
     3. treat it like a diary
     4. if a milestone was reached: git tag
  5. back to 2
  6. before ending a day push (git push)

Visual Studio

preparing for the seminars/ programming:

1. Install and check dotnet (dotnet –version) should be 5.XXX
2. Check and install git cmd client (git --version) (github desktop is not enough)

3. DON'T SUBMIT without checking that you did not modify the wrong files
    1. Git status is your friend!!!

Visual Studio

Stepping through the example project

Useful commands:

- dotnet new tool-manifest
- dotnet tool install Cake.Tool –version 0.38.4
- dotnet cake --target=Build
- dotnet cake --target=Tests
- dotnet new console –o <Name>
- dotnet new sln –n <Name>
- dotnet sln add <project>

Useful files/folder:

- Cake.build
- .config
- tools

Computational Thinking/ Algorithmic Thinking

Abstraction:

- Looking at different levels/layers of the problem
- extracting smaller essential pieces
- Ignore pieces of problem that are not crucial
- Solving the pieces
- Integrate them and repeat

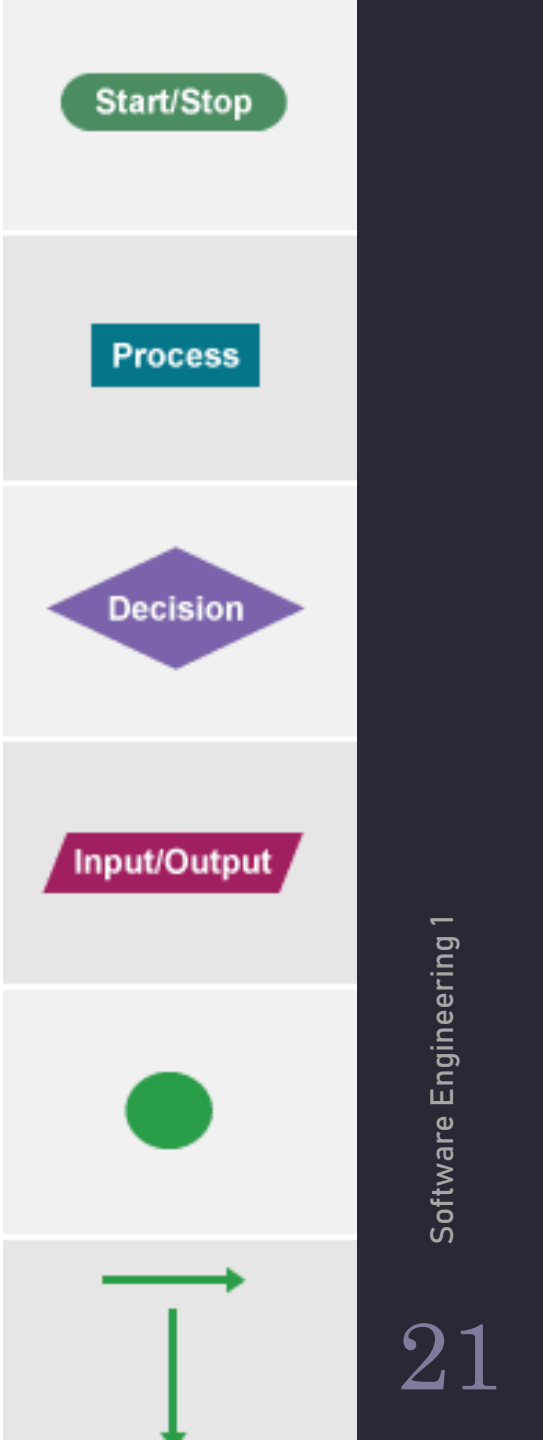
» Creates model of the actual problem

Flow Chart Diagrams

Name	Symbol	Usage
Start or Stop		The beginning and end points in the sequence.
Process		An instruction or a command.
Decision		A decision, either yes or no.
Input or Output		An input is data received by a computer. An output is a signal or data sent from a computer.
Connector		A jump from one point in the sequence to another.
Direction of flow		Connects the symbols. The arrow shows the direction of flow of instructions.

PseudoCode

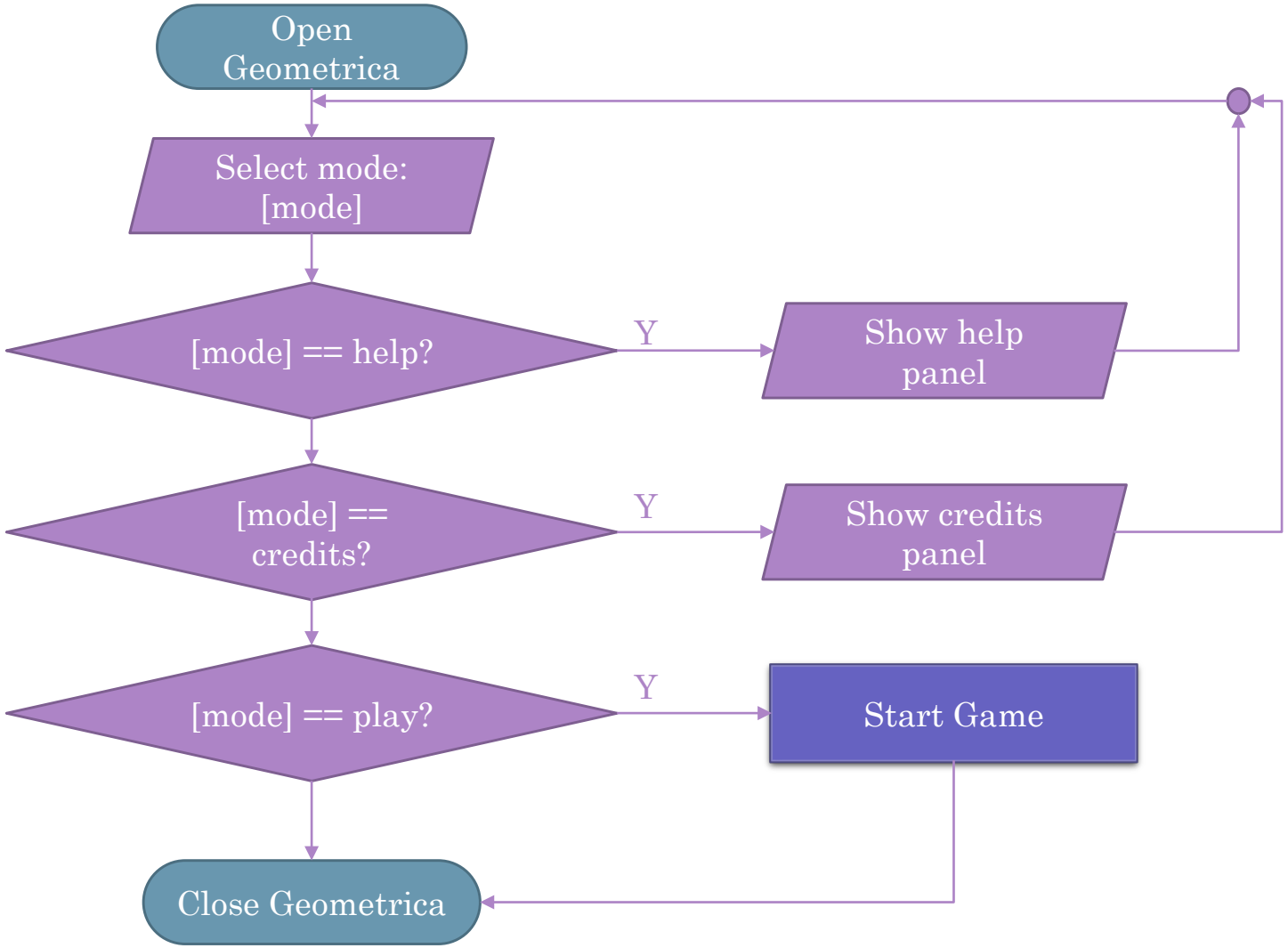
- Expressing processes
- without visual representation
- Elements:
  - Input
  - Output
  - While
  - For
  - Repeat-until
  - If-then-else
  - Specific instructions
- ordered top to bottom



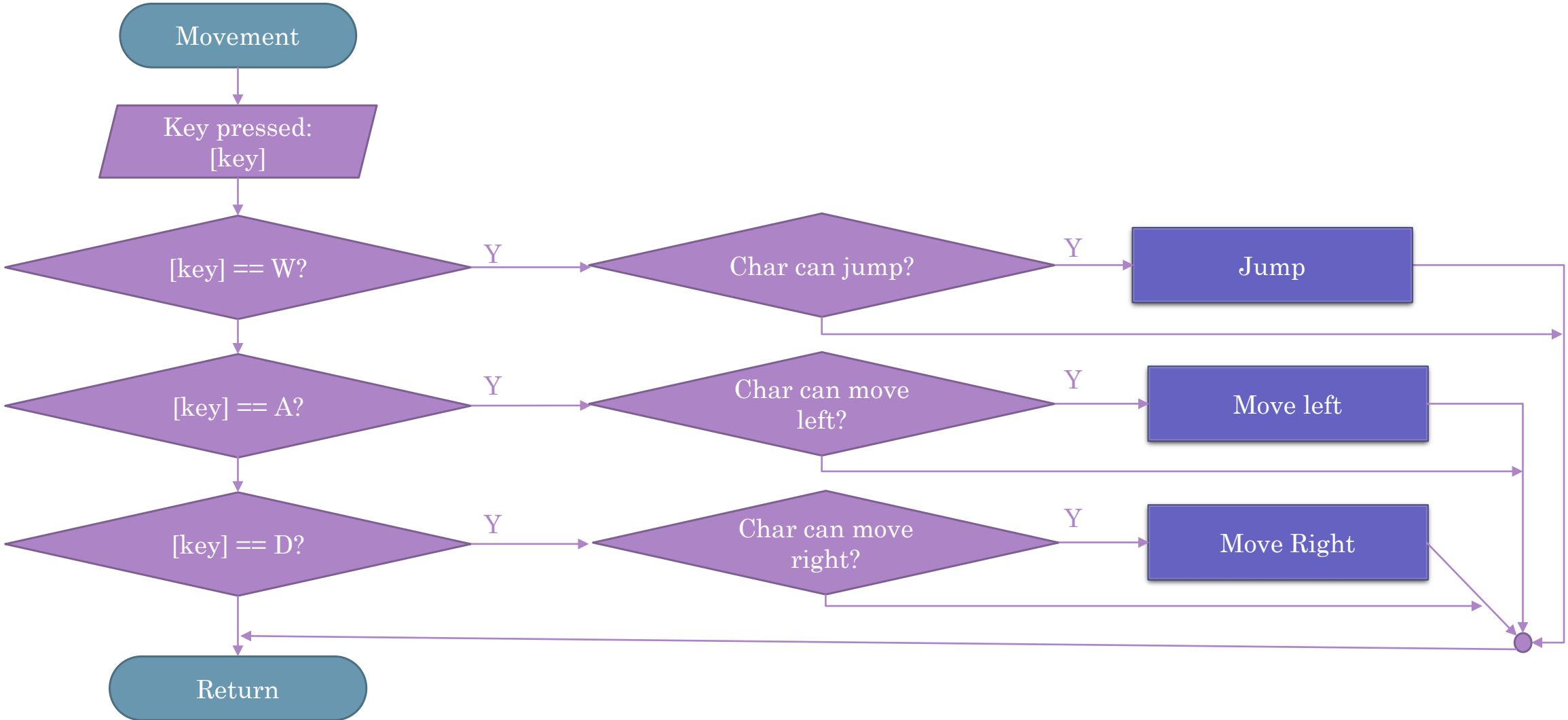
PseudoCode

- Take the flowchart from Geometrica
- Translate the process into pseudocode
- Discuss in your teams the outcome

PseudoCode

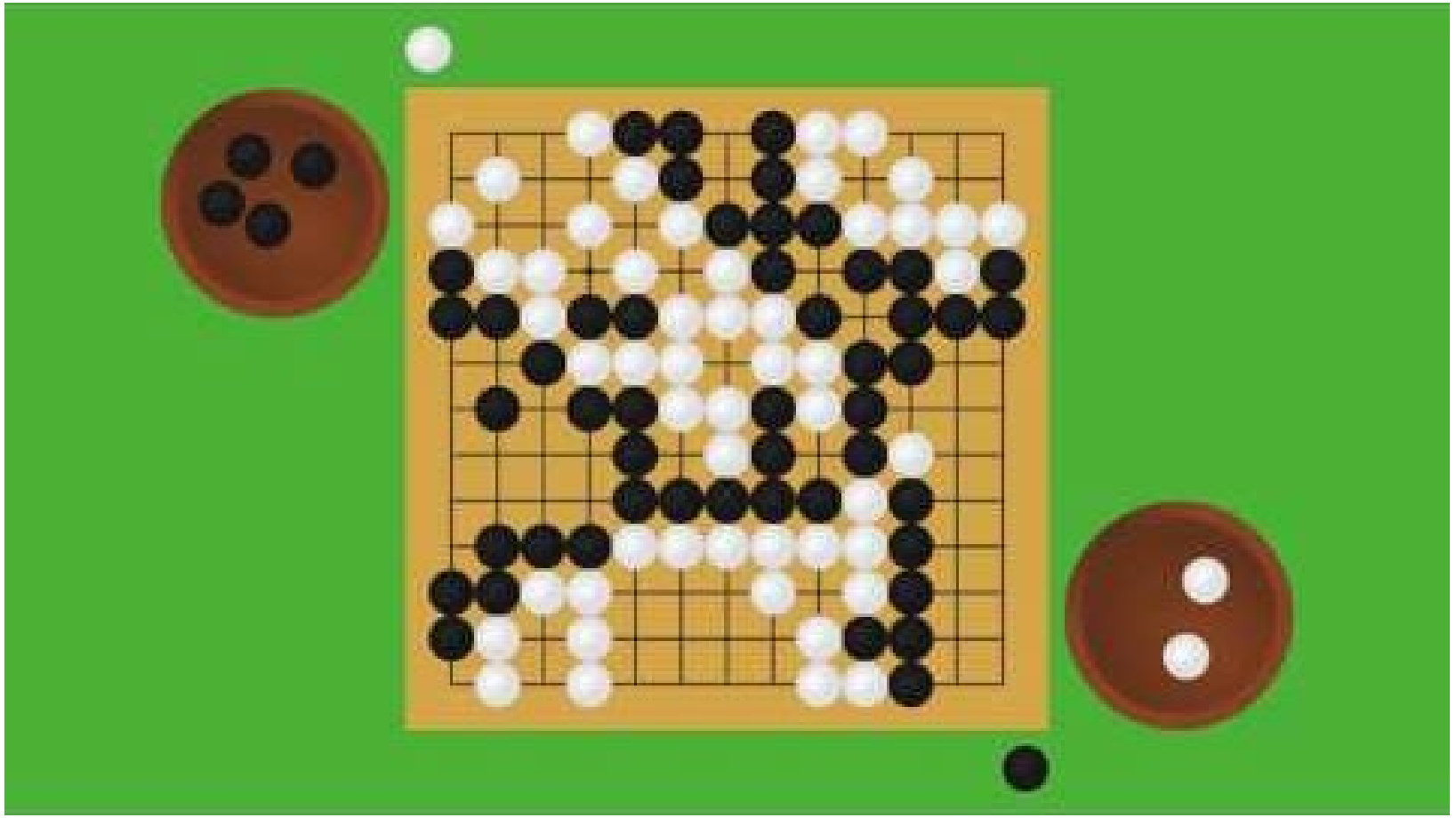

Open Geometrica
Select mode: [mode]
[mode] == help?
Y
Show help panel
[mode] == credits?
Y
Show credits panel
[mode] == play?
Y
Start Game
Close Geometrica

PseudoCode


Movement
Key pressed:
[key]
[key] == W?
Y
Char can jump?
Y
Jump
[key] == A?
Y
Char can move
left?
Y
Move left
[key] == D?
Y
Char can move
right?
Y
Move Right
Return

PseudoCode (Self-Study Exercise)

- Write down the rules and then comparing them:



https://www.youtube.com/watch?v=5PTXdR8hLlQ