

# Vertical Slice Assignment - April 20th

## Overview

The goal is having an executable that proves the core gameplay of the project. It should be presented to the publisher to prove that core gameplay mechanics are good enough to create a larger game.

## Content

Delivery is basically a **playable version 0.5** that will be presented in class to verify the game will be fun and profitable. It is also a test to verify the team is capable of delivering working software.

This delivery must include the following working elements from the provided checklist:

### General

- ☐ Screens transitions: fade-in/fade-out or better transitions
- ☐ GOD mode (F10): no-damage, collisions display, increase resources, spawn any type of unit, kill units,...
- ☐ Game resolution: 1280x720px@60fps, support for windowed mode and fullscreen
- ☐ Automatic Continuous Integration and Deployment

### Logo Scene

- ☐ Logo implemented: custom art, avoid plain text (no video necessary yet)
- ☐ AUDIO: Sound FX characteristic for the logo

### Title Scene

- ☐ Main game title implemented, custom art, avoid plain text
- ☐ AUDIO: Sound FX characteristic for the title appearing
- ☐ UI: Main game menu implemented: START - EXIT (minimum, add more options as convenient)
- ☐ UI: AUDIO: Buttons/UI fx feedback sounds

### Gameplay Scene

- ☐ Gameplay:
  - ☐ Implement a first version of the map
  - ☐ Implement at least 2 types of Resources
  - ☐ Implement at least 2 types of Playable Units (gatherer, combat) with basic functionality (Movement + 1 action)
  - ☐ Implement group selection and movement
  - ☐ Implement at least 2 types of Buildings (Base center + extra)
  - ☐ Buildings with some **basic** functionality (create units, upgrade/boost)
  - ☐ Implement clear win/lose conditions

☐ HUD:

☐ Minimal HUD implementation depending on the game:

☐ Units/building creation bars

☐ Units/building improvements (research, level up, ...)

☐ Extra HUD functionality from core features of your game

☐ UI: Minimap: Minimal functionality: Map image and camera movement (units do not need to be displayed yet)

☐ UI: ANIM: Interface elements should be animated with visual feedback

☐ UI: AUDIO: Sound feedback for the interface elements (when required)

☐ AUDIO: Minimum of 10 fx game sounds implemented

☐ AUDIO: Music track specific for the screen

☐ AUDIO: Positioning in the screen: Volume effects (sounds too far away from the screen should not be heard)

**Win/Lose Scene**

☐ Custom background with victory/lose messages (custom font)

☐ AUDIO: Music track specific for the screen

## Submission Rules

Each team **MUST** upload their **release build** as a **zip** file to the folder “*Vertical Slice*” on the campus website **no later than Monday April the 20th at 23:59**. If more than one file is uploaded, only the last one will be evaluated. The build **MUST also** be published in the **Release section of the project’s GitHub page**.

### Release folder structure and naming conventions:

```
> Team_Name-Game_Name_v0.5.zip      // Game directory zipped
    Game_Name                        // Game name directory
        Assets                       // Assets directory, it could contain
                                    // multiple sub-dirs and files
        Game_name.exe                // Main binary for the game
        Xxx.dll                      // All required DLLs to run the game
    LICENSE.md                       // Game license file
    README.md                        // Game detailed info
```

NOTE: Additionally, GitHub release **MUST** contain detailed information on the current release (new features, improvements...)

Submission **will not** be accepted for grading in case:

- It is not delivered on time
- Build is malformed (files that are not needed or not compiled in Release mode)
- It does not work on university’s computers (regardless if it works somewhere else)
- Build is not available **also** in the GitHub Release system
- Game does crash while testing (should be able to play to win/lose the game)
- Team Members not present during presentation will be graded “NP”

## Grading Criteria

- **100% Checklist points:** Evaluation will consider all points completed from the checklist. Note that gameplay-related points will be evaluated with priority.