

Alpha Assignment - May 18th

Overview

The goal is having an executable that is fully playable with all gameplay bits of your game. The game will be presented to the publisher to be accepted as a *valid Alpha* ready to enter *Beta phase*.

Content

Delivery is basically a **playable version 0.8** 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 any missing item from *Vertical Slice checklist* plus 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
- ☐ ANIM: Animated Logo in/out (fading, movement...)

Title Scene

- ✓ Main game title implemented, custom art, avoid plain text
- ☐ Load/Save game options implemented and functional
- ✓ 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
- ☐ UI: Main game menu implemented: START - CONTINUE - OPTIONS - EXIT
- ☐ UI: ANIM: Animated menu, responsive, visual feedback...
- ☐ ANIM: Title animation with easings (fading, movement...)
- ☐ AUDIO: Music track specific for the screen

Options Scene

- ☐ UI: Basic game config options implemented with custom font
- ☐ UI: OPTION1: Fullscreen mode selection
- ☐ UI: OPTION2: Music volume slider
- ☐ UI: OPTION3: Fx sounds volume slider
- ☐ ANIM: Animated options menu, responsive, visual feedback...
- ☐ AUDIO: Sound feedback for buttons and sliders
- ☐ AUDIO: Music track specific for the screen

Gameplay Scene

✓ Gameplay:

- ☐ **Implement all gameplay-specific features that your projects require**
- ✓ Implement a first version of the map
- ✓ Implement at least 2 types of Resources
- ☐ Implement at least 3 types of Resources
- ✓ Implement at least 2 types of Playable Units (gatherer, combat) with basic functionality (Movement + 1 action)
- ☐ Implement at least 4 types of Playable Units
- ✓ 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
- ☐ Implement fog of war
- ☐ Implement a (short) tutorial to guide the player through the mechanics
- ☐ Implement at least 3 Quests/Missions (can be related to the tutorial)
- ☐ Implement at least 4 Dialogs, it could be between characters
- ☐ Implement at least 1 particle effect
- ☐ Implement AI behaviour for the units (it can be updated later)

✓ 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: Minimap with fog of war
- ✓ UI: ANIM: Interface elements should be animated with visual feedback
- ✓ UI: AUDIO: Sound feedback for the interface elements (when required)
- ☐ UI: PAUSE menu available on pause
- ☐ UI: PAUSE menu animated with feedback (custom font)
- ☐ AUDIO: PAUSE menu in/out animation with audio feedback

- ✓ AUDIO: Minimum of 10 fx game sounds implemented
- ☐ AUDIO: Minimum of 15 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)
- ☐ AUDIO: Music effects fade-in/fade-out, lower volume on PAUSE

Win/Lose Scene

- ✓ Custom background with victory/lose messages (custom font)
- ✓ AUDIO: Music track specific for the screen
- ☐ ANIM: In/out animations (easings) for ending messages
- ☐ AUDIO: Sounds for win/lose messages and animations

Submission Rules

Each team **MUST** upload their **release build** as a **zip** file to the folder “*Alpha*” on the campus website **no later than Monday May the 18th 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_v0.8.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.