P1 - Alpha Assignment - June 14

Overview

Your goal is to finish an exact copy of your arcade game for specific amount of content.

Submission

Each team should upload a **link to their project website on github.com** inside the folder “Alpha” of the campus website before **June 14 at 23:59**. The work will be presented in the Project class by each team (if an individual does not show up to the presentation he/she will have a “NP”). No need to create any presentation. The build must **also** be uploaded to github under the Releases section.
The release 1.0 should be a **zip** containing:

- **README.md**:
    - Link to the github.com page of the project
    - Llist of team members and github accounts
    - Short description of the game
    - How to play the game, detailing the controls
    - Any additional comments for the teacher to understand some part of the project

- **LICENCE.md**:
    - Chose a licence that fits your project - https://choosealicense.com/

- **Game files:**
    - Executable compiled in Release with all necessary DLL files
    - A resource *folder* with all the media files (png, wav, ogg)
    - *No other file must be there!* - Be sure to remove any code and unnecessary files
    - Maximum of **20 Mb** of zipped build

- **Tasks list (pdf file):**
    - Lists of tasks delivered by each team member
    - (optional) Initial estimation time + real development time.
    - (optional) Each task should have a delivery date.

- **Website:**
    - Website using a template in github.com
    - Short description of the game, names and a team photo
    - Short description of the main contributions of each team member
    - **Embed** gameplay video of ~1/2 min presenting the purpose of the project, the gameplay and the debugging features.
    - Link to **directly** download the latest release

The folder structure should be the following:

```
> TeamName_GameName_v.X
    > GameName
        - Assets/Resources
        - .dll files
        - GameName.exe
    - README.md
    - LICENCE.md
    - Licences (all third party libraries' licences)
    - Tasks list
```

Functionality list

All projects must try to replicate the original arcade game to the maximum detail. Specific levels or features might be requested depending on the project. Once those levels are completed by the player, the game must jump to the game's ending screen. All the original game features should be included in the project unless specifically discarded in the following list. Setting menus and highscores need not to be included in any game.

This is a **minimum** funcionallity list, any extra features are encouraged.

General

- The game must have an initial screen with:
    - Description of the project
    - Subject, degree, university
    - Team members
    - Others: webpage, twitter, team logo,...
- The game must be playable in a loop without closing the executable
    - Welcome screen -> Gameplay screen -> Win / Lose -> Welcome Screen
- All musics and audio effects should be added as in the original game.
- The game plays in desktop full scren and uses gamepad as main input method.
    - It must still be playable with keyboard
- Minimum debugging functionality in order to test the game properly
    - God mode, collision boxes, direct win/lose…

U.N. Squadron

- The game must contain 2 levels (fully implemented, including bosses)
    - N° 2 already implemented for the vertical slice
    - Another level of your choice
- Shop screen should be implemented
    - The total money can be maintained through different gameplay loops
    - Pick your choice from a minimum of 4 weapon types
    - Buying new airplanes does not need to be included
- Power-Ups (Pow and Total) should be implemented
- Basic debugging functionality should include changing weapon, infinite ammo, infinite money,...

Donkey Kong

- Pick a minimum of 3 levels
    - Stage 4 was already implemented for the vertical slice
    - Try to implement all of them to the maximum detail!
- Basic debug functionallity should include spawning an enemy at mouse location using a key

Pengo

- Implement the full game as far as you can reach
- Levels do not need to be generated randomly
- Basic debug functionality should include jumping to a specific level, spawning enemies and blocks at mouse location using a key,...

Pang

- Implement at least the first 6 levels
- The game demo at the beginning is not necessary
- Power-ups and lifes should be fully implemented
- Animals and foood can be discarded
- Basic debug functionality should include switching weapon types, spawning a power-up or a ball at mouse location using a key

Snow bros

- Implement the first stage (10 levels) with all enemies and its final boss
- The tickets loot and the bonus phase can be discarded
- Power-ups can be discarded, but they are encouraged as the game might be harder to win otherwise
- Basic debug functionality should include spawning an enemy at mouse location using a key

Acceptance Criteria

To accept the submission it must meet the following rules:

- The game is delivered on time.
- The game does not crash while testing in university's computers (a test to achieve win/lose).
- Using STL libraries is forbidden.
- Website contains description, **direct** link to download executable and a ~1/2 min video.
- Build is correct (only the necessary files, .exe compiled in Release, correct folder structure).
    - An incorrect build will have a penalization of 50% of the grade

Grading Criteria

The submission will only be accepted if they follow the rules from the *Acceptance Criteria* section.

- **Features (70%)**: All requested features are implemented and working correctly.
- **Polish (30%)**: Graphical coherence, fx implemented, fluid animations and gameplay and everything else that mimics the original game. Extra debug functionality will also be taken into account as polish.

Previous years webpage examples

- https://wilhelman.github.io/Gunbird_P01/
- http://carcasanchez.github.io/Puzzle-Bobble/
- http://trodek.github.io/Project-1-Team-SUMHC-Pro-Name/