Videogame project delivery notes

Videogames Technology
Asignatura transversal

Departamento de Automática

Objectives

- Clarify how to deliver the videogame project
- Point out some practical issues

Bibliography

None

Project deliverables

Project deliverables

- Videogame source code
- Design document: How the game will be
- Project report: What has been done
- Instructions

Delivery method → GitHub

- Source code in the repository
- Documentation also in the repository

Evaluation procedure

After the exam, the instructor will

1. Fork all the repositories
2. Download the project from GitHub
3. Read instructions
4. Try to execute the project
5. Read the documentation and source code
6. Inspect GitHub activity
7. Set individual and group califications

Evaluation platform:

- PC with Ubuntu

Evaluation criteria

- Technical quality (group) - 20 %
  - 0 if the videogame does not execute properly
- Game design (group) - 20 %
  - Game originality will have a big impact in this criteria
- Documentation quality (group) - 10 %
- Teamwork (individual) - 20 %
  - Activity in GitHub and coevaluation will have a big impact in this criteria

Source code structure

Recommended source code structure

- `src/` or `<project name>`: Source code
- `docs/`: Documentation
  - `docs/gdd.md`: Game Design Document
  - `docs/memoria.md`: Project memory
- `tests/`: Unitary tests (not used in our project)
- `dist/`: Distribution (not used in our project)
- `resources/`:
  - `resources/characters`:
  - `resources/data`:
  - `resources/maps`
  - `resources/tilesets`
  - `resources/sounds`
- `README.md`: Instructions

In general, always try to keep your code neat and nice

Memory contents

- Team members and roles. Do not include missing people
- How the team has been organized
- Degree of the GDD that has been accomplished
- Explain why the GDD has not been fully accomplished, if it applies
- **Technical aspects of the project you want to be considered**
- Any additional information that the instructor should know

Remember CCC: context, content, conclussions

Typical problems (I)

Paths

Use relative paths, ALWAYS

Path separator

Use proper path separator

Windows style:

- `resources\sprites\alien.png`

Unix style (Linux and MacOS):

- `resources/sprites/alien.png`

Python trick: `os.path.sep`

- `spritePath = 'resources' + os.path.sep + 'sprites'`

Linux is case sensitive! Use capital letters in paths with care

Typical problems (II)

Extra trick:

- Any relative path depends on the working directory
- Solution: change working directory

```
file_path = os.path.dirname(os.path.abspath(__file__))
os.chdir(file_path)
```

Not everybody has a RTX 3090TI ...

- Use a reasonable resolution

Not all the files have been included in the repository

- Proper testing should avoid this risk

The only branch that will be considered for evaluation is 'master'

- Your code in that branch must be fully operational

Last remarks

Remember to test the game properly ...

- ... and ideally using an issue tracker

Python app deployment: `https://wiki.python.org/moin/deployment`

Game presentation

One presentation per group

- 5-7 minutes per group
- Use any resource at will

CCC

Context
Content
Conclussions