PA5 – Documentation

Student Information

Integrity Policy: All university integrity and class syllabus policies have been followed. I have neither given, nor received, nor have I tolerated others' use of unauthorized aid.

I understand and followed these policies: (•) Yes ( ) No

Name: Joey Domino

Date: 3/10/22

Submission Details

Final ***Changelist*** number: 137666

Verified build: (•) Yes ( ) No

Required Configurations: does not apply

YouTubeLink: does not apply

Discussion (What did you learn):

Any questions about the documentation here....

I tried to make my document feel a bit like a story. "Here's my big problem (how to make the army) and here's how I did it. As I explain building out the army, I bring up problems that occurred and how I solved them. Hope that works!

Verify Builds

- Follow the Piazza procedure on submission
  - Verify your submission compiles and works at the changelist number.
- Verify that only MINIMUM files are submitted
  - No – Generated files
    - *.pdb, *.suo, *.sdf, *.user, *.obj, *.exe, *.log, *.pdb, *.db, *.user
    - Anything that is generated by the compiler should not be included
  - No – Generated directories
    - /Debug, /Release, /Log, /ipch, /.vs
- Typical files project files that are required
  - *.sln, *.csproj, *.cs,
  - App.config, AssemblyInfo.cs, CleanMe.bat
  - Resources Directory:
    - *.tga, *.dll, *.wav, *.glsl, *.azul

Standard Rules

Submit multiple times to Perforce

- Submit your work as you go to perforce several times (at least 5)
  - As soon as you get something working, submit to perforce
  - Have reasonable check-in comments
    - Points will be deducted if minimum is not reached

Submission Report

- Fill out the submission Report
  - No report, no grade

Code and project needs to compile and run

- Make sure that your program compiles and runs
  - Warning level 4
  - NO Warnings or ERRORS
    - Your code should be squeaky clean.
  - Code needs to work “as-is”.
    - No modifications to files or deleting files necessary to compile or run.
  - All your code must compile from perforce with no modifications.
    - Otherwise it's a 0, no exceptions

Project needs to run to completion

- If it crashes for any reason…
  - It will not be graded and you get a 0

No Containers

- Containers (No automatic containers or arrays
- Template or generic parameters
- No arrays
  - You need to do this the old fashion way - ***YOU EARNED IT***

Leave Project Settings

- Do NOT change the project or warning level
  - Any changing of level or suppression of warnings is an integrity issue

Simple C#

- No .Net
- We are using the basics
  - Types:
    - Class, Structs, intrinsic types (int, float, bool, etc…)
  - Basics language features
    - Inheritance, methods, abstract, virtual, etc…

No Debug code or files disabled

- Make sure the program has only active code
  - If you added debug code or commented out code,
    - please return to code to active state or remove it

Adding files to this project

- Make sure you add the files in the appropriate sub-directories
- Make sure any new files are successfully integrated into the project
- Make sure your new files are submitted to Perforce

Due Dates

- See Piazza for due date and time
- Submit program perforce in your student directory assignment supplied.
- Fill out your this ***Submission Report*** and commit to perforce
  - ***ONLY*** use Adobe Reader to fill out form, all others will be rejected.
  - Fill out the form and discussion for full credit.

Goals

- Is your design document good?
    - Get some feedback now… before its due

Assignments

Grading:

- ***Design Paper:***
    - Write a ***2 FULL pages*** - paper design pattern for feedback
    - See description below

Required Features:

Paper: Design Document... suggestions

- Just make sure each problem/pattern answers a set of questions

For example:

A. ***State a challenge*** in our Space Invader's Game (a paragraph)
    - Problem:
        - We had the problem of moving the objects together. …. blah blah blah
    - Solution:
        - A collection of objects needed to move in unison, to treat the 55 aliens as a collection I used a Composite pattern

B. ***Pattern description***..... (1-2 paragraph)
    - Intent - what does the pattern solve... why is it useful
    - UML - simplified (don't show too much) of the pattern YOU used...
        - (not from the internet)
    - Use class view (export as PNG, and cut and paste that into document)
        - Keep it small 1/2 page
        - Do not show _ALL_ the variables or the complete UML of the game
        - Only the portion that is related to the pattern
    - Need to see YOUR pattern the one actually used in your application
        - Not the Internet's
        - The UML from Visual Studio is good enough

C. Key Object Oriented mechanics
    - How this pattern accomplishes the job is.... (1-2 paragraphs)
    - What's the Object Orient mechanics...
        - How is the accomplished

D. ***How it is used in Space Invaders***.... (1-2 paragraphs)
    - Describe the uses in the space invaders game...

- Talk to me in the document
    - Pretend you are speaking to me

This is for you…

Create Diagrams

- UML for all the systems
    - Doesn't have to be perfect UMLs
    - the class view diagram in visual Studio is good
- Use the UML to prototype

Development

- Store project in student directory in perforce
- Place your PDF in the PA5 directory
    - Work on the document here.
        - Call it SampleDesignDoc.pdf

Submission

- Submit your PA5 directory into perforce:
    - /student/<yourname>/PA5/...
- Fill out the Submission report and submit that pdf to your student directory

Validation

Simple checklist to make sure that everything is submitted correctly

- DNA

Hints

Most assignments will have hints in a section like this.

- DNA

Troubleshooting

- DNA