Zariba Game Academy

OOP & Quality Code Teamwork

Teamwork Specification

General

You have to create your own MonoGame simple Game. **YOU ARE NOT ALLOWED TO MAKE BOARD GAMES! - e.g. Ludo, Chess, Double Mill etc.**

Requirements

1. **You MUST USE a Source Control System - TFS, GitHub, SourceTree, TortoiseGit, CodePlex etc.**
2. Each team has to choose a **Project Manager** who will present the game in front of jury. The responsibility of the Project Manager is to help with all aspects of the game development - programming, graphic design, game design, giving out the tasks and keeping the deadlines for the game. The project manager will preferably (but not mandatory) be the most experienced developer in the team. He will be helping with coding the most complicated parts of the game.
3. Choose a team name.
4. You need to create a **brief** (2-3 pages) **GDD** which should include the following information:
   1. Presentation of the team - names and roles
   2. Brief description of the game idea
   3. Graphic Design implementation
   4. Implementation and algorithms used
   5. Design Patterns
   6. OOP structure
   7. Future plans and ideas for the game - new modes, levels, functionality, characters,... as per your idea.
5. **DO NOT PLAGIARIZE** - do not copy paste your games from the internet. Try to create your own simple game. It does not have to be original - e.g. Flappy Bird or a variation of Flappy Bird will be a fine idea, given that you do not copy paste it from the internet. If you do copy, we will know and the whole team will be disqualified from further modules in the course.
6. **A completely finished game is required.**

Programming and Game-specific requirements

1. **StyleCop should yield 0 warnings.**
2. **YOU MUST USE A SOURCE CONTROL SYSTEM!**
3. **YOU MUST NOT MAKE A BOARD GAME!**
4. **Follow all guidelines for Quality Code, as in the lectures (40% of total score).**
5. **Use at least 3 Design Patterns (as in the Monopoly Game and others).**
6. **You must use State Machine Design Patterns.**
7. **MVC pattern is optional for 10%.**
8. **Use Abstract Classes, Interfaces, Inheritance, Classes, Constructors, Lambda etc.**
9. **Make a completely finished game!!! - each bug, or not implemented functionality will be sanctioned.**
10. **You must have the following States Intro, Main Menu, Game, Pause, Game Over. You can add others if you want.**
11. **Additional points will be awarded for functionality, complexity, game idea and so on.**

Presentation and Deliverables

1. You will have 10-15 minutes to present your game in front of the judges.
2. You will be using the equipment in the lecture theatre.
3. You should bring your Game and GDD on a flash drive and copy it on the PC.
4. You can make a short Power Point presentation to help you present your idea (Optional).

Good Luck and Have Fun.