COMP397 – Web Game Programming

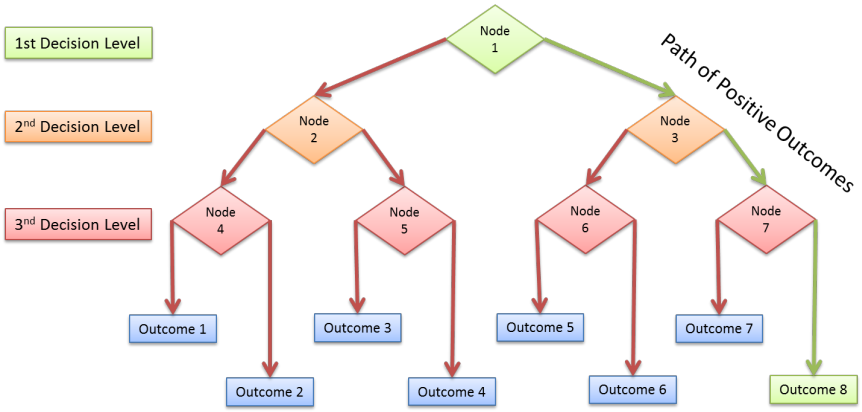
Assignment 1
Document the Dragon

Due Week #4 (Friday February 5, 2015) @ midnight.

Value 10%

The Dragon Game **Maximum Mark: 52**

Overview: Use "The Dragon Game" site files provided on GitHub and example code from class to build your project – a choose-your-own-adventure Web Game. You will use the **State Machine Design Pattern** described in class to create each Scene of your application. Create both an internal and external document for the site. Ensure you follow documentation best practices and use the external documentation template provided. Currently the site uses **impress.js** and only gives the player the choice of exploring cave 1 or cave 2 (or one overall decision level or node). You will create your own story and extend the functionality of the site with the create.js suite. Your game will include 2 more decision levels for the player. **Example:** after going into the cave 1, the player is given the option to turn right or left (2nd decision level). He chooses left and proceeds to the "inner cave" where he can choose to fight the dragon or try and convince the dragon to spare his life (3rd decision level).



Instructions :

(18 Marks: Functionality, 12 Marks: Internal Documentation, 4 Marks: Site Structure, 10 Marks: External Documentation, 4 Marks: Version Control, 4 Marks: Cloud Deployment)

1. Add two decision levels to the game for a total of 8 possible outcomes (see decision tree diagram above) **(18 Marks: Functionality):**
    a. Each Decision Level will allow the player at least 2 choices. (12 Marks: Functionality).
    b. Only one positive outcome is possible for the player. (2 Marks: Functionality)

    c. Use the **State Machine Design Pattern** described in class to create separate **scenes** for each story node (4 Marks: Functionality)
    d. *Option:* You may allow the player to go back to the previous node or cross laterally to a node on the same level
    e. *Option:* You may allow the player to find "special items" to help him win.
2. Include **Internal Documentation** for your program and create an appropriate **site structure** for your project **(12 Marks: Internal Documentation, 4 Marks: Site Structure):**
    a. Ensure you include a program header that indicates: The Source file name, Author's name, Last Modified by, Date last Modified, Program description, Revision History (6 Marks: Documentation).
    b. Ensure your program uses contextual variable names that help make the program human-readable (2 Marks: Documentation).
    c. Ensure you include **inline comments** that describe the conditional structure you will use for your decision tree matrix (4 Marks: Documentation)
    d. Ensure your project files are organized within an appropriate site Structure, separating the **View** (HTML Document and CSS Files) and your **Game Logic** (JavaScript and TypeScript files) (2 Marks: Site Structure).
    e. Use **TypeScript** to create a scalable, object-oriented application (2 Marks: Site Structure).
3. Include **External Documentation** for your program that includes **(10 Marks: External Documentation)**:
    a. A company Logo (2 Marks: External Documentation).
    b. Table of Contents (2 Marks: External Documentation).
    c. Version History (2 Marks: External Documentation).
    d. Detailed Game Description – describing each node and any possible outcomes (4 Marks: External Documentation)
4. Share your files on **GitHub** and deploy to a **Cloud Service** (Microsoft Azure, Heroku, etc.) to demonstrate Version Control Best Practices **(4 Marks: Version Control, 4 Marks: Cloud Deployment).**
    a. Your repository must include **your code** and be well structured (2 Marks: Version Control).
    b. Your repository must include **commits** that demonstrates the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).
    c. Ensure your game is live and online. Deploy to a Cloud Service of your choice (4 Marks: Cloud Deployment).

SUBMITTING YOUR WORK

Your submission should include:

1. An external document (MS Word or PDF).
2. A link to your project files on GitHub.
3. A link to your live site on a Cloud Service of you choice.

This assignment is weighted **10%** of your total mark for this course.

Late submissions:

- 20% deducted for the first day late.
- 10% deducted for each additional day.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.