Games Studio 1

*Introduction and Requirements*

**Introduction**

Neverwinter Nights has been widely recognized as one of the best computer games implementations of the pen-and-paper *Dungeons and Dragons* RPG game mainly because of its *Aurora toolset*, enabling game players to become *builders*, e.g., the creators behind modules and adventures.

After the release of Neverwinter Nights 2 builders can use the improved *Electron toolset*, which is even more user-friendly and flexible than Aurora.

The user community behind NWN 1&2 is still thriving while producing modules, whose quality can often rival the official modules by Bioware and Obsidian.

Neverwinter Nights mods can enter the *IGF* (Independent Game Festival) modding competition at the *Game Developers Conference* (the others being UT2004 and Half-Life 2).

**General Requirements**

Students will be required to join a team, generally comprising 3-4 members, including programmers, designers and digital artists.

Exceptions to this requirement are possible where specific conditions apply (e.g., a team cannot find a digital artist to join their team), and a different marking scheme will be used to evaluate their efforts.

The goal of the assignments in Games Studio 1 is designing and implementing a Neverwinter Nights 2 module.

Particular emphasis will be given to the ability of teams to translate their design documents into *functional, creative* and *well documented* implementations.

The final submission (at the end of semester one teaching period) will include both a complete design document and an archive containing a NWN2 module plus associated resources (if any).

Each team is supposed to complete the design document in three incremental stages, as outlined in the roadmap below.

**Assignment 1:** Due 23 March 2007 (3 p.m.)
**Assignment 2:** Due 27 April 2007 (3 p.m.)
**Assignment 3:** Due 1 June 2007 (3 p.m.)

The 1st assignment deals with the *conceptual design* stage; the 2nd assignment is dedicated to the *implementation of a prototype*, whereas the 3rd assignment gives rise to the *final implementation and testing*.

Further information is reported below where crucial activities are listed, the level of completion required per each stage (assignment), and who should be responsible to carry them out.

Clearly, *D* designates *Designers*; *P* indicates *Programmers*, and finally *A* means *Artists*.

Whenever a role is indicated in parentheses, the involvement of the student covering that role is usually considered optional, but encouraged nonetheless.
<table>
<thead>
<tr>
<th>Activity</th>
<th>Stage#1</th>
<th>Stage#2</th>
<th>Stage#3</th>
<th>Who does what</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Setting and back-story</td>
<td>Draft</td>
<td>Complete</td>
<td>Complete</td>
<td>D</td>
</tr>
<tr>
<td>2. Characters (+ bios)</td>
<td>Draft</td>
<td>Complete</td>
<td>Complete</td>
<td>A (D)</td>
</tr>
<tr>
<td>3. Flow chart of game events</td>
<td>Draft</td>
<td>Complete</td>
<td>Complete</td>
<td>P (D)</td>
</tr>
<tr>
<td>4. Game areas (plot)</td>
<td>List</td>
<td>Complete</td>
<td>Complete</td>
<td>D</td>
</tr>
<tr>
<td>5. Rough map</td>
<td>List</td>
<td>Complete</td>
<td>Complete</td>
<td>A</td>
</tr>
<tr>
<td>6. Scripts</td>
<td>List</td>
<td>Complete</td>
<td>Complete</td>
<td>P</td>
</tr>
<tr>
<td>7. Conversations</td>
<td>List</td>
<td>Complete</td>
<td>Complete</td>
<td>D</td>
</tr>
<tr>
<td>8. Custom content</td>
<td>List</td>
<td>Complete</td>
<td>Complete</td>
<td>A (P)</td>
</tr>
<tr>
<td>9. Concept Art (1, 2, 4, 5, 7, 8, 9)</td>
<td>List</td>
<td>Complete</td>
<td>Complete</td>
<td>A</td>
</tr>
</tbody>
</table>

**Draft** indicates a preliminary implementation; **list** requires to incorporate a list of assets (e.g., scripts, areas, etc.) in the documentation, whilst **complete** indicates a 100% level of completion for a certain activity.

**Marking**

The following examples are provided just as a rough guideline: Specific examples might not correspond exactly to students’ marks because of factors not detailed in this outline.

**HD-level** module: A complete NWN2 module, featuring multiple populated areas, NPCs scripting & conversations. The quality of presentations and of the documentation produced will be **flawless**. Features not examined during the tutes will be present in such a module.

**DI-level** module: A complete NWN2 module, featuring multiple populated areas, NPCs scripting & conversations. The quality of presentations and of the documentation produced will be **good**. Features not examined during the tutes are **NOT** expected in such a module. Besides, some technical flaws might be present in the module that will prevent it to be HD-level (e.g., gameplay issues, bugs in the scripting of NPCs, etc.).

**CR-level** module: A complete NWN2 module, featuring multiple populated areas, NPCs scripting & conversations. The quality of presentations and of the documentation produced will be **sufficient**. Features not examined during the tutes are **NOT** expected in such a module. Besides, some technical flaws might be present in the module that will prevent it to be DI-level (e.g., gameplay issues, bugs in the scripting of NPCs, etc.).
**PA-level** module: A NWN2 module, featuring multiple populated areas, NPCs scripting & conversations but not necessarily complete. The quality of presentations and of the documentation produced will be just **sufficient**. Features not examined during the tutes are **NOT** expected in such a module. Besides, some technical flaws will be present in the module that will prevent it to be CR-level (e.g., serious gameplay issues, bugs in the scripting of NPCs, etc.).

**NN-level** module: The module will suffer from severe lack of features and/or bad implementation. It is recommended not to aim at such a result.