

**DXC Codes 2018 Scoring Criteria**

There are 5 criteria that are each worth 10 points, making 50 the highest possible score. For any criteria, a score of 8 points is considered "excellent", meaning the highest likely score is around 40 points, with 9-10 point evaluations reserved only for projects that merit special commendation

**1-3 Points**

**4-6 Points**

**7-8 Points**

**9-10 Points**  
Extra credit for exceptional quality

<p><b>Criteria 1: Game Aesthetic</b></p> <p>Does the project use good artwork and good sound design? The art and sound should be consistent with the overall aesthetic of the game. The artwork includes the backdrop, the sprites, and the animation of those things. The sound design includes the sound effects and also any music.</p>	<p>Art and sound are either low quality, largely missing, inconsistent with the aesthetic of the game, or makes use of previously published works that have not been acknowledged by the project</p>	<p>The quality, consistency, or quantity of art or sound is uneven. <b>E.g.:</b> the artwork is quite good, but the sound design is not. Or both the artwork and sound design are good, but they are not consistent with the theme of the project</p>	<p>Art and sound design both meet all of the following criteria: high quality, self-consistent, ample, and either (a) the material is acknowledged by the project (in the case of previously published material) or (b) the material is original to the project (e.g., artwork, sound effects, or music created by students specifically for this project)</p>	<p>Aesthetic design that is truly inspired</p>
<p><b>Criteria 2: Software Design and Implementation</b></p> <p>Does the project use an appropriate amount of well-written, well-documented scripts?</p> <p>"Appropriate amount" of scripting does not necessarily mean "huge amounts" of scripting. The intent is not to give awards to the projects that have the most code. Rather, the intent is to give awards to projects that did enough coding to make the game perform well.</p> <p><b>Pay special attention to:</b></p> <ul style="list-style-type: none"> <li>• Sprite movements that are smooth</li> <li>• Game controls that are responsive</li> <li>• Automated opponents that behave intelligently</li> <li>• Game physics that behaves consistently</li> <li>• Scripts that are well documented</li> </ul>	<p>Software performs poorly and is poorly documented</p>	<p>The software quality is uneven. Some things about the project's coding are good, but some things about the coding are not good. <b>E.g.:</b> there are copious scripts, but they do not perform well (e.g., sprite control is shaky or sluggish, the physics behaves inconsistently, the automated opponents are "stupid", etc.). Or there are only a few scripts, but they're poorly documented.</p>	<p>Code meets all of the following criteria: ample amounts of high quality code, that performs well, and is well documented</p>	<p>Software design that is truly inspired</p>
<p><b>Criteria 3: Software Quality (Bugginess)</b></p> <p>Is the project relatively bug-free?</p>	<p>Software is either (a) very buggy, or (b) so minimal that the idea of bugs isn't even applicable. <b>E.g.:</b> a project that has almost no scripts and no interactivity should be considered buggy, as should a project with many scripts that are so prone to unintended performance that the game is essentially unusable</p>	<p>Software bugs do not unduly impact the playability of the project. The user notices minor bugs that for the most part don't detract from the overall enjoyment of the project</p>	<p>Ample amount of code that appears to be largely bug-free</p>	<p>Complex software design that has somehow avoided defects despite its complexity</p>
<p><b>Criteria 4: CLEAR Values</b></p> <p>Does the project reflect DXC's CLEAR values? There are many ways to score points in this category:</p> <ul style="list-style-type: none"> <li>• Project is just plain fun, and players naturally want to return to it for their enjoyment</li> <li>• Project is wonderfully interactive, with many interesting things for the player to do</li> <li>• Project is ambitious, and impresses the player with the extent of its scope</li> <li>• Project is could be held up as an example to other projects, of how to execute a project well</li> <li>• Project is original, and not derivative of past projects.</li> </ul> <p>Criterion graded on 10-point scale, where 8 points means "very excellent". Scores of 9-10 are rare, and only for exceptional quality.</p> <p><b>Note that a project does not have to do all of these things to score well in this category.</b></p>	<p>Little or no CLEAR values displayed</p>	<p>Some CLEAR values displayed, and some of them are displayed well</p>	<p>Many CLEAR values displayed, and they are displayed well</p>	<p>Extra credit (for example, all CLEAR values are displayed very well).</p>
<p><b>Criteria 5: Clear Instructions</b></p> <p>Are the instructions sufficient to make playing the project frustration-free? Some examples:</p> <ul style="list-style-type: none"> <li>• Makes use of up-front animations to explain gameplay. (Extra credit if the up-front animations are skippable for repeat players – remember the goal is to be frustration-free).</li> <li>• Makes use of an in-game Help button so that players can review instructions at any time.</li> <li>• Makes use of just-in-time instructions, so that players are told how to operate a part of the project just before they get to that part.</li> <li>• Includes written instructions on its main projectpage.</li> </ul>	<p>Instructions are low quality, confusing, incomplete, or are largely missing altogether</p>	<p>Instructions are "good enough" for the player to figure out how to operate the project, with only a bit of frustration. Mid-level scores should also apply for projects that have "too much" instruction – e.g., the instructions are so long that it's difficult to find the truly necessary information</p>	<p>Instructions are high quality, complete, and completely frustration free</p>	<p>Instructions are as entertaining as the project itself</p>