nscc

Game Programming: Portfolio Requirements

Application Review Process:

Applications are reviewed in the order they are received. Applications meeting the criteria of program prerequisites and portfolio review will be offered a seat in a program section, if available. Accordingly, early preparation and submission is recommended. If your portfolio is not successful, deficiencies will be noted and you will be given the opportunity to address them and re-submit your application, as time permits.

Portfolio Assessment:

Learners in the Game Programming program devote many hours to learning programming, game engines, and game development. The development of programming skills enables them to code methods, classes, systems, managers, programming patterns, and architectures. These skills will be used to design and code games within a pre-existing game engine and from scratch. The admissions committee will look at your portfolio for evidence of your potential to reach this goal and to be successful.

Portfolio Submission

Once completed, you may send your digital portfolio through one of the following methods:

- If sharing through a link/URL, send an email to: Admissions@nscc.ca
    - Please include your name, your W-number, the program you are applying to, and the link in the body of the email.
    - Verify that the link is active and accessible from any computer.
    - Ensure files are well-organized; reflect the names of the portfolio components; and can be easily located, opened, and read by computers other than the one you used to create your portfolio.
- If submitting digitally:
    - Acceptable file formats:
        - Documents:
            - .pdf
            - Microsoft Word document
            - .rtf
            - .txt
        - Images:
            - .jpg/.jpeg
            - .gif
            - .png
        - Games / Demos / Projects:
            - .zip
            - .rar
- If submitting as attachments to an email, send to: Admissions@nscc.ca

- The maximum size of the email cannot exceed 25Mb
    - Please include your name, your W-number, and the program you are applying to in the body of the email.
    - Ensure files are attached to the email; well-organized; reflect the names of the portfolio components; and can be easily located, opened, and read by computers other than the one you used to create your portfolio.
- If submitting on a memory stick:
    - Please include your name, your W-number and the program you are applying to with your submission.
    - Ensure files are well-organized; reflect the names of the portfolio components; and can be easily located, opened, and read by computers other than the one you used to create your portfolio.
    - Send to:
      NSCC Admissions
      PO Box 220
      Halifax, NS B3J 2M4
    - Alternatively, you can bring your memory stick to any NSCC campus, and it will be forwarded to the Admissions Department

Portfolio Options

We offer two portfolio options for your application, perform one of the following options below

Option #1 - Personal/Original Work:

Requirements:

Submit three (3) examples of code that demonstrate your strongest programming work (from any programming language).

For each example provide the following:

- A cropped screenshot focused on a section of your code you want to present.
  (You do not need to send the entire script, just show us a section that you feel best demonstrates your coding skill.)
- A detailed explanation of what that code does.

Your examples should collectively demonstrate the following introductory programming abilities:

- Variables
- operators (+, -, *, /, =, ==, …)
- methods (calling of)
- conditionals (if-else)
- loops
- basic algorithms (code should perform an action)
- logic principles (translation of simple logic ideas into code)

Checklist: *(Option #1)*

Completed	Option #1 Components
	Three (3) cropped screenshots of code samples
	Document explaining in detail what each code sample does

Rubric: (Option #1)

	Incomplete: 0pts	Novice: 60pts	Intermediate: 80pts	Advanced: 100pts	Results:
Programming	Code examples and explanations provided do not demonstrate an adequate understanding of coding fundamentals, or insufficient code examples provided.	Code examples and explanations provided demonstrated a basic understanding of coding fundaments.	Code examples and explanations provided demonstrated an understanding code at an intermediate level.	Code examples and explanations provided demonstrated an understanding code at an advanced level.	/100

Option #2 – Logic and Programming Exercise:

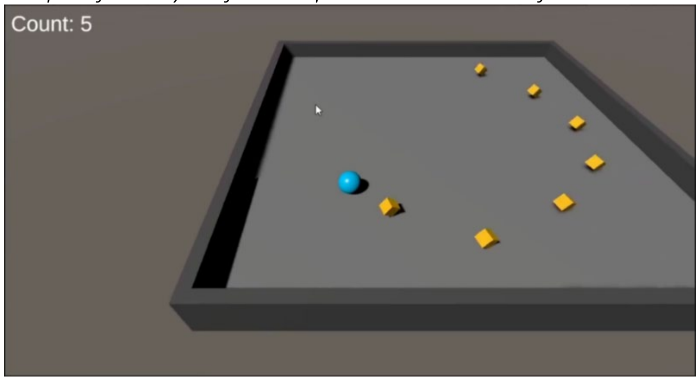
Complete all of the following:

1. Roll-a-Ball Project:

Perform the following:

1. Download and install the free-to-use Unity software:
    - https://unity.com/download
2. Complete Unity's Roll-a-Ball Tutorial:
    - https://learn.unity.com/project/roll-a-ball
    - The committee is looking for ability to follow technical instructions and demonstrate introductory programming ability which includes problem solving. You are allowed to use the Unity Forums: https://forum.unity.com/

Example of what your final output should look like after the tutorial:



3. Apply your knowledge:
    - Apply the knowledge you have learned in the tutorial by expanding it further. The committee is looking for you to demonstrate a high degree of comprehension and application of the tutorial content. Demonstrate your comprehension by modifying it or adding new content to the project.
    - Suggestions:
        i. You learned how to create objects to make your play area (floors and walls). Try to add more objects to your project. Make a more interesting level, add obstacles, new collectibles, etc.
        ii. You learned how to use triggers to collect items. Try using them to trigger something else.
        iii. You learned how to make the ball move. Try making it move differently. Try making something else move.
        iv. You learned how to add UI elements. Try adding additional UI screens, text, or information.
        v. You learned to use many of the tools/techniques in Unity. Try using those tools to add something new or different to your project, to demonstrate your comprehension.

Clarification: The above suggestions are based off of knowledge that would be learned in the Roll-a-Ball tutorial.

2.Reflection Document: *(Option #2)*

Create a document where you outline your thought process throughout the tutorial and application of your learning. Add what you feel is necessary. Your thoughts should be reflective and not simply descriptive of what you did. Make sure you at least cover the following:

- What went right?
- What problems/challenges did you face? Did you solve them? How?
- Where did you look for answers?
- What addition did you make to demonstrate your comprehension?

Submission Requirements: *(Option #2)*

Submit the following:

- Your build folder (labelled "Builds" in the tutorial – subject to change) and all its contents, as a compressed archive .zip or .rar. (The files and folders found inside your build folder is the playable build of your Roll-a-Ball project.)

  NOTE: Do not just send us your single project executable (.exe) file. If your compressed folder does not include all the required files your project will not run. We recommend copying your compressed folder onto another computer and testing your build before submitting.

- Source code files (.cs files) – include all C# source files you created.
- Reflection document.

Checklist: *(Option #2)*

Completed	Option #1 Components
	Roll-a-Ball build folder (compressed as .zip or .rar file)
	C# source code files (.cs) you created
	Reflection document

Rubric: (Option #2)

	Incomplete: 0/80 pts	Novice: 48/80 pts	Intermediate: 64/80 pts	Advanced: 80/80 pts	Results:
Roll-a-Ball Project	Project is incomplete or is not complete to a level that demonstrates introductory competency in use of game programming tools. Did not demonstrate ability to follow detailed, technical instructions, or no evidence of ability to understand introductory programming concepts or did not complete the tutorial. Did not demonstrate comprehension of tutorial content. No application of knowledge, by modifying or adding new content.	Project is complete at a minimal / basic level and demonstrates some competency in use of game programming tools. Demonstrated ability to follow detailed, technical instructions, ability to understand introductory programming concepts, and completed the tutorial. Demonstrated comprehension of tutorial content in a minimal manner. Basic application of knowledge, by modifying or adding new content.	Project is complete at an intermediate level and demonstrates good competency in use of game programming tools. Demonstrated ability to follow detailed, technical instructions, ability to understand introductory programming concepts, and completed the tutorial. Demonstrated comprehension of tutorial content in a clear and distinct manner. Significant application of knowledge, by modifying or adding new content.	Project is complete at an advanced level and demonstrates exceptional competency in use of game programming tools. Demonstrated ability to follow detailed, technical instructions, ability to understand introductory programming concepts, and completed the tutorial. Demonstrated exceptional ability to comprehend and expand upon tutorial content. Significant and advanced application of knowledge, by modifying or adding new content.	/80
	Incomplete: 0/20 pts	Developing: 12/20 pts	Competent: 16/20 pts	Exemplary: 20/20 pts	

Reflection Document	No reflection document or reflection document was very poorly formatted and/or contained several grammatical errors.	Simplistic observations. Provides little or no insight. Comments are more descriptive than reflective.	Adequate degree of observations. Some insight and analysis on work completed.	Sophisticated and thoughtful observations. High degree of insight and analysis on work completed.	/20
Total:					/100