CS-223
CODE REVIEW DOCUMENT

for

Project 3
Virtual Tour Based Game

Prepared by:
Group-20
Inderpreet Singh Chera - 160101035
Shubehndu Patidar - 160101068
Siddharth Sharma - 160101071

April 22, 2018

Contents

1 Introduction

The intent of a code review is to catch bugs/issues/defects before the offending code is deployed to a production environment and to transfer knowledge of implementation details to the rest of the team. Code review involves a slow code inspection phase to check errors.

1.1 Goals

The main reasons for performing code review is to :

- Finding bugs, since bug finding in code review are easier to find and fix, than later in testing.
- Adherence to coding conventions
- Improving code quality and understandability
- Increasing efficiency, by finding trivial programming errors like data resource wastage or use of uninitialized variables.

1.2 Code Inspection

The aim of code inspection is to discover some common types of errors caused due to oversight and improper programming practice. In this phase, team members are selected and asked to perform the inspection, in which we check the coding standards. Then, we look for the presence of certain kinds of errors like modifying a formal parameter while the routine calls a constant parameter. Considering the statistics some common programming errors which are to be checked are :

- Use of uninitialized variables
- Jumps into loops
- Improper storage allocation and deallocation
- Incompatible assignments
- Non terminating loops
- Array indices out of bounds

- Mismatches between actual and formal parameter in procedure calls
- Use of incorrect logical operators or incorrect precedence among operators
- Improper modification of loop variables
- Comparison of equality of floating point variables

All these are checked and reported. It is important to note that code inspection is a slow phase and no more than 400 lines of code are to be checked at a time. The report is then given to the author for appropriate changes to be done.

2 Coding Guidelines/Task Description

- Declaration of Global Constant Variables should be done in All Caps.
- For Class variable naming lowerCamelCase is used.
- Local variables are named all small and are separated by underscore.
- Class names are in CamelCase.
- Function naming is done in CamelCase.
- **Allman** indentation style is used for curly braces.
- 4 spaces indentation is used.
- Variables are to be named meaningfully.
- Headers are absolutely necessary in each module.
- Comments are required for each function.
- On an average four lines of code should have a single line of comment.
- Do not use a coding style that is too clever or too difficult to understand.
- Avoid obscure side effects: The side effects of a function call include modification of parameters passed by reference, modification of global variables, and I/O operations. An obscure side effect is one that is not obvious from a casual examination of the code.

- Header Format is as follows :-

  <header>

  Module Name

  Date of creation

  Author

  Modification history

  Synopsis

  Functions

  Global variables accessed/ modified

  < /header>

3 Code Testing Team

3.1 Team Profile

The code testing team comprises of the following members, all of whom are Undergraduates currently pursuing Bachelor of Technology at Indian Institute of Technology Guwahati, India in the Department of Computer Science and Engineering. All of the members are currently in the sophomore year.

- Mohit Singh
- Sparsh Bansal
- Ritik Agrawal

All members of the team are proficient in C# and have past experience in developing games through unity3D.

4 Code Inspection Reports

4.1 Code Report by Mohit Singh:

- In script database.cs variable name initialPos is not meaningful
- In script database.cs variable named temp and i is not descriptive enough
- Indentation is proper throughout the scripts
- In all scripts Headers are informative and descriptive
- No loops are non terminating
- There are no goto statements used
- There are no uninitialized variables found
- Most of the variables name are informative
- All the scripts are well commented

4.2 Code Report by Sparsh Bansal:

- Jump (go to) statements were not found.
- All loops will terminate according to their stopping conditions.
- Function name is in lower camel casing. For example : line 54 in DisplayScore.cs.
- Proper headers of each module are provided with details about Name of Module, Date of Creation, Synopsis, Functions and Global variables.
- Some variable names example, pos and toggle, could have been more descriptive
- Unused import packages are not there. were found.
- Proper commenting is done with summaries for every function.
- Bounded lists are used.
- Code is too crowded at some places which reduces readability of the code. For example: Visualizer.cs.
- Uninitialised and unused variables are not found.
- Function names are proper and quite descriptive.

4.3 Code Report by Ritik Agrawal:

- CamelCase is not consistent among the functions name
- In the visualizer.cs script a different function could have been made to load the database.
- Functions are too long in some scripts.
- In visualizer.cs variable named pos is not intuitive
- Proper indentation is done
- In QuestionAnswers.cs variable cur_value is not intuitive.
- Headers are descriptive and properly formatted.
- All loops are terminating
- A variable named temp is used in visualizer.cs which Is meaningless
- In questionAnswers.cs toggle named variable is used which is also the name of a gameobject which is quite confusing.
- No goto statements are used in the scripts.
- Almost all variable names are descriptive .

5 Conclusion

1. Some functions length were to large.
2. Some variable names were not clear(are not intuitive).
3. Overall coding style is good.