



Abstract

This research focuses on the development of a game theme-based educational module that helps novice students understand object-oriented programming (OOP) concepts, specifically, encapsulation, polymorphism, and inheritance. This effort is motivated by the popularity and effectiveness of educational games. The module was implemented using Python within Vizard, a virtual reality development toolkit. Three-dimensional (3D) models featured in the module were created in 3ds Max 2014, a 3D modeling and animation software application. The results demonstrate that the module helped students learn OOP concepts and found the module user-friendly.

Problem Statement

- Research has shown that the gaming approach to education makes concepts easy to learn and understand as well as engaging to students.
- The concepts covered in the module are:
 - Inheritance – when a child class inherits attributes (functions and variables) from its parent class
 - Encapsulation – making certain attributes accessible from another class by assigning three different protection levels:
 - Public: the attribute is accessible from every class
 - Private: the attribute is accessible from only other attributes within the same class
 - Protected: the attribute is accessible from child classes
 - Polymorphism – creating different versions of a function, variable, or object

Methodology

- Module development involved the Constructivist Theory which states that a person “builds” knowledge of a concept from his/her own experience of that concept.
- Student engagement and interactivity were also considered during module implementation.

Implementation

- Implementation was completed in three phases:
 - 3D Models were developed in 3ds Max
 - The models were imported into a format compliant with Vizard
 - Functionalities were added in Vizard using Python

Module Overview

An executable (.exe) file is generated from the Python source code. Once a user clicks on the executable file, he or she is first greeted with a main menu where three games on either inheritance, polymorphism, or encapsulation can be chosen. Each game starts with a tutorial on either one of the three topics.



Main Menu

- User is greeted with an overview of the module and is presented with three games to choose from.

Inheritance Game

- In this game, the user builds either a land or air vehicle.
- As the vehicle is built, a UML diagram is modified to display vehicle type (air or land) and purpose (commercial, military, or private).

Polymorphism Game

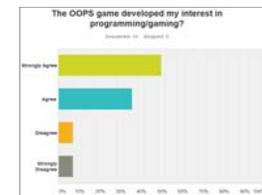
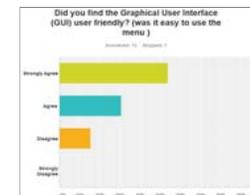
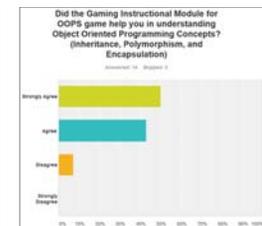
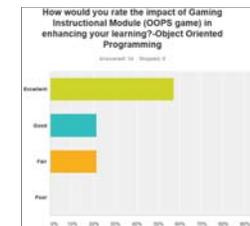
- The user faces off against the computer in a guessing game where either the computer or the user must correctly guess a number between 1 and 20.

Encapsulation Game

- Here, the user takes a quiz where he/she must correctly assign a protection level (public, private, or protected) for a given variable.
- If the user is willing, he/she can take another quiz asking him or her on the accessibility of a variable in another class.

User Study & Results

- During the Fall 2014 semester, the module was evaluated with 14 undergraduate students in a programming course.
- The students experimented with the module and took a survey afterwards.
- Survey data demonstrates that the module makes a significant educational impact on students.



Conclusion

- Future work includes conducting additional user studies and developing other instructional modules on linked lists, stacks and queues, and trees.
- The educational module helps students learn OOP concepts and is easy to use making it enjoyable to students.

Acknowledgements

This study was supported in part by National Science Foundation (NSF), HBCU-UP (TIP), Award Number: HRD-1137541 and HRD-1238784. The authors would like to thank the program manager, Dr. Claudia M. Rankins for the support.