Course Outcomes Guide #4

Course Title: SDE 205 Game Programming II

Course Leader: David Maruszewski

Expected Learning Outcomes for Course

- Demonstrate principles of video game production
- Create event driven programming environments and algorithms
- Develop game events through the use of C++
- Define and demonstrate the differences in programming and developing of different genre games
- Analyze, select and apply tools appropriate for a specific solution
- Logically formulate scripts and/or programs to solve problems
- Understand and articulate interactivity in the gaming industry, including the connectivity between computer art and programming
- Apply programming and artistic theory in practical applications
- Demonstrate problem solving skills through verbal and written media
- Apply rudimentary Physics and Trigonometry principles
- Address simulation and gaming solutions with professionalism and ethics

Assessment
(How do students demonstrate achievement of these outcomes?)
Students are required to complete their second project which was created to test skills gained throughout the course. The project is then graded with a "grade sheet" which looks at skills and outcomes. A full assessment rubric may be created in the future.

A supplemental (3rd of the semester) exam is issued to help confirm the findings of the project grade.

Validation
(What methods are used to validate your assessment?)
Currently, all grades sheets are held for two semesters and composite data is used to show trends. When this course reaches three sections per semester, a spreadsheet will be created in order to track outcomes. The exams should help verify or contradict findings.

Results
(What does the data show?)

1. Students can manage coding in project form
2. Students have a hard time retaining syntax in coding
3. Students can pick up programming interactivity relatively easy
4. Error trapping, and creating flawless algorithms is challenging to the student
5. Debugging for some students is very easy, but others seem to be difficult. The split is about 50/50.
6. Being able to explain their programming is something students do not do well. They don't give results. They give commentary.

Follow-up
(How have you used the data to improve student learning?)

Course Outcomes Guide #4

1. Projects are to be kept and improved every semester
2. Time in class is used to look at this
3. This is encouraged through team projects and in class work. We also changed 205 to be a true continuance of SDE 104 (Game Programming I)
4. Students will be put in teams in order to help them with debugging and error trapping.
5. I pair up stronger students with weaker students to help them out. This can be beneficial to both if done correctly.
6. More hands-on teaching will be applied when dealing with report writing. Results will be stressed.

Budget Justification

(What resources are necessary to improve student learning?)

1. A game engine is used in this class and helps to get the students interested in programming issues.
2. Thawspace would be beneficial for downloading of Open Source material.
3. This was run as a Hybrid, which allowed students to do work from home. Webinar software would work well in this course.