

ITEC 2270: Application Development Documentation

Jackie Harmon

2024-05-10

Contents

1	Course Overview	3
2	Evolution of Course Timeline	3
2.1	Original Timeline	3
2.2	Adjusted Timeline	3
2.3	Future Timeline	4
2.4	All Timelines Gantt Chart	4
3	Course Structure	5
3.1	Step 1: Demonstration Activity 1 (DA 1)	5
3.2	Step 2: Demonstration Activity 2 (DA 2)	5
3.3	Step 3: Demonstration Activity 3 (DA 3)	6
3.4	Step 4: Demonstration Activity 4 (DA 4)	6
3.5	Step 5: Final Project	7
4	Evaluation of Course Assessment Criteria	8
5	Lessons Learned and Improvements	8
5.1	DA 1 Sprint:	8
5.2	DA 2 Sprint:	8
5.3	DA 2 Code Review:	8
5.4	DA 3 Code Review:	8
5.5	DA 4 Code Review:	8
5.6	Reflective Assignment:	8
5.7	Timeline:	8
5.8	Workshops:	9
6	Course Materials	9
6.1	Instructional Videos	9
6.2	Answer Keys	9
6.3	Workshops	10
6.4	Textbook	10
6.5	Example Game	10
7	Project Management	10
7.1	Planning	10
7.2	Execution	10
8	Evaluation and Feedback	10
9	Conclusion	11

1 Course Overview

Course Title: ITEC 2270 - Application Development

Instructor: Professor Jackie Harmon

Description: In this course, students modify an arcade or tabletop game using Python and SQL. The course is structured into five steps, including four Demonstration Activities (DAs) and a Final project. Each step is supported by instructional “how-to” videos. Students submit their code for feedback and create videos showcasing their work, discussing challenges, and highlighting features added. Additionally, the course demonstrates how to break a large project into sprints, with each DA representing a sprint and the Final being the project end.

Course Assessment:

Programming Assignments 1% overall bonus, 8 PA = 8% overall total bonus

Demonstration Assignments 48%, 4 DA = 12% each

Review Questions 15%, 5 RQ = 3% each

Reflective Assignment 5%

Final 32%

2 Evolution of Course Timeline

This course project spans 17 weeks.

2.1 Original Timeline

This timeline was initially planned for the course:

- **Week 1:** Entry Quiz
- **Week 2:** DA 1
- **Week 3:** Review and Debugging
- **Weeks 4-5:** DA 2
- **Week 6:** Review and Debugging
- **Weeks 7-8:** DA 3
- **Week 9:** Review and Debugging
- **Week 10-12:** DA 4 (Week 11 Spring Break)
- **Week 13:** Review and Debugging
- **Week 14:** Final Project
- **Week 15:** Review Others’ Games
- **Week 16:** Reflective Assignment
- **Week 17:** Contingency Week

2.2 Adjusted Timeline

The following adjustments were made based on student feedback and observed challenges:

- **Week 1:** Entry Quiz
- **Week 2:** DA 1
- **Week 3:** Review and Debugging
- **Weeks 4-5:** DA 2
- **Week 6:** Review and Debugging
- **Week 7:** Project Outline (added for scalability understanding)
- **Weeks 8-9:** DA 3

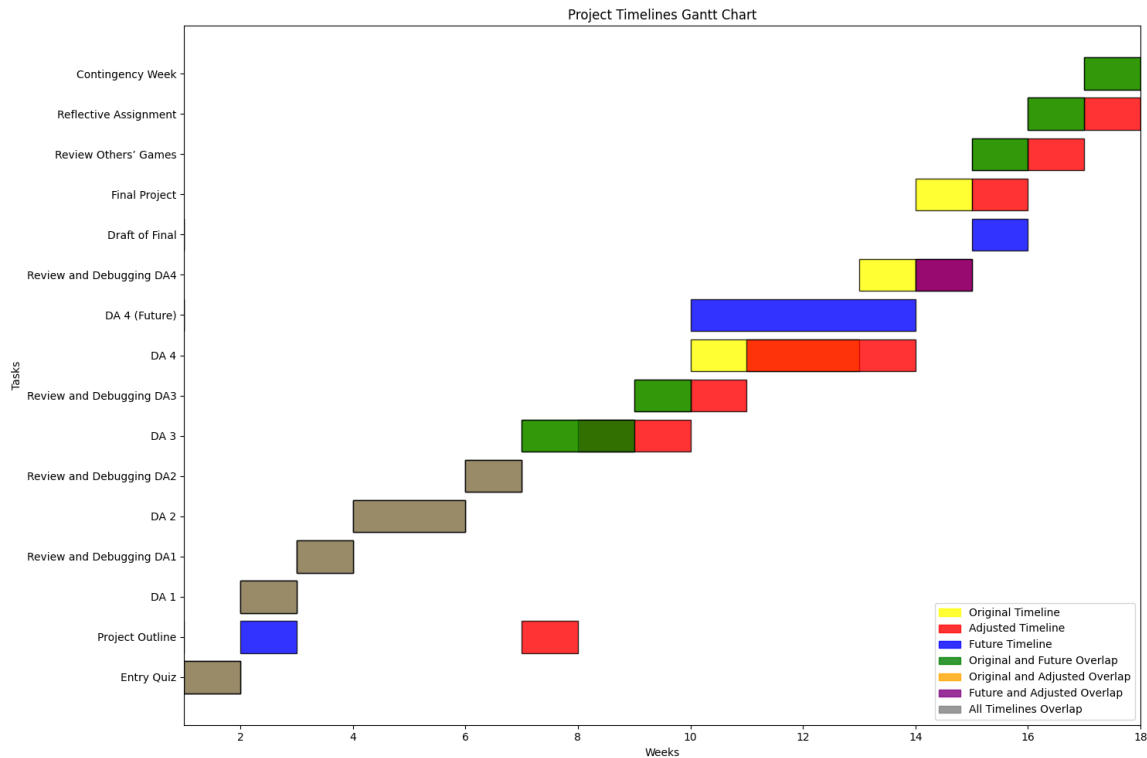
- **Week 10:** Review and Debugging
- **Weeks 11-13:** DA 4 (Week 11 Spring Break)
- **Week 14:** Review and Debugging
- **Week 15:** Final Project
- **Week 16:** Review Others' Games
- **Week 17:** Reflective Assignment

2.3 Future Timeline

Based on the experience and feedback from this course iteration, the following timeline is proposed for future courses:

- **Week 1:** Entry Quiz
- **Week 2:** DA 1 & Project Outline (Added for scalability understanding)
- **Week 3:** Review and Debugging
- **Weeks 4-5:** DA 2
- **Week 6:** Review and Debugging
- **Weeks 7-8:** DA 3
- **Week 9:** Review and Debugging
- **Week 10-13:** DA 4 (Adjusted from 3 weeks to 4; week 11 Spring Break)
- **Week 14:** Review and Debugging
- **Week 15:** Draft of Final & Review Others' Games (Added for additional testing)
- **Week 16:** Final Project & Reflective Assignment (Pushed back for additional debugging)
- **Week 17:** Contingency Week

2.4 All Timelines Gantt Chart



3 Course Structure

3.1 Step 1: Demonstration Activity 1 (DA 1)

- **Objective:** Grab code from an existing game and demonstrate how it works.
- **Example Game:** Snake
- **Python Libraries:** `turtle`, `time`, `random`
- **Prerequisite Knowledge:** Chapters 3-9 from ITEC 2260 - Introduction to Computer Programming.
- **Resources Provided:**
 - Lecture videos with examples for each chapter (combined chapters 5 & 6).
 - Instructional video using the Snake game to demonstrate requirements.
 - Workshop: Python Beginners (Coming from C#).
- **Student Deliverables:**
 - **DA 1:**
 - * Assignment Submission: Submit code for feedback.
 - * Discussion Forum: Create a 2-4 minute video showcasing the game and discussing any issues encountered and features added.
 - **Bonus Assignment:**
 - * Programming Assignment 1 (Chapter 9 #9)

3.2 Step 2: Demonstration Activity 2 (DA 2)

- **Main Objectives:**
 - Display scores.
 - Keep high scores between games using shelving (alternatively, can use pickling).
- **Additional Objectives:**
 - Add fun elements (sound effects, replayability, background images, sprites).
- **Example Game:** Snake
- **Python Libraries:** `pygame`, `Vector2` from `pygame.math`, `sys`, `shelve` or `pickle`
- **Prerequisite Knowledge:** Chapters 10 (Classes & Object-Oriented Programming) & 11 (Inheritance) from the Python textbook by Gaddis.
- **Resources Provided:**
 - Lecture videos covering chapters 10 & 11.
 - Instructional video using the Snake game to demonstrate requirements.
 - Workshop: Making the Project Scalable.
- **Additional Learning:**
 - Convert code to Object-Oriented Programming.
- **Student Deliverables:**
 - **DA 2:**
 - * Assignment Submission: Submit code for feedback.
 - * Discussion Forum: Create a 2-4 minute video showcasing the game and discussing any issues encountered and features added.

- **Additional Assignments:**
 - * Review Questions for Chapters 10 & 11
- **Bonus Assignments:**
 - * Programming Assignment 2 (Chapter 10 #9)
 - * Programming Assignment 3 (Chapter 10 #7)

- **Check-In:**

- Students provide feedback on their chosen game and discuss its suitability for the course requirements. They have the option to change their game if necessary. Project Outline creation added.

3.3 Step 3: Demonstration Activity 3 (DA 3)

- **Main Objective:** Create in-game status features (leveling up, progress bar, timer).
- **Additional Objectives:**
 - Add menus (main/start menu, pause menu, game over/play again menu).
 - Add more fun elements (sound effects for leveling up and game end, background music, out of bounds area, hide cursor during play).
- **Example Game:** Snake
- **Python Libraries:** pygame_gui
- **Prerequisite Knowledge:** Chapters 12 (Recursion) & 13 (GUI Programming) from the Python textbook by Gaddis.
- **Resources Provided:**
 - Lecture videos covering chapters 12 & 13.
 - Instructional video using the Snake game to demonstrate requirements.
- **Student Deliverables:**
 - **DA 3:**
 - * Assignment Submission: Submit code for feedback.
 - * Discussion Forum: Create a 2-4 minute video showcasing the game and discussing any issues encountered and features added.
 - **Additional Assignments:**
 - * Review Questions for Chapters 12 & 13
 - **Bonus Assignments:**
 - * Programming Assignment 4 (Chapter 12 #4)
 - * Programming Assignment 5 (Chapter 12 #2)
 - * Programming Assignment 6 (Chapter 13 #8)
 - * Programming Assignment 7 (Chapter 13 #6)

3.4 Step 4: Demonstration Activity 4 (DA 4)

- **Objective:** Create a database to make a scoreboard displaying the top 3 usernames and scores using sqlite3.
- **Example Game:** Snake
- **Python Libraries:** sqlite3

- **Prerequisite Knowledge:** Chapter 14 (Database Programming) from the Python textbook by Gaddis.
- **Resources Provided:**
 - Lecture videos covering chapter 14.
 - Instructional video using the Snake game to demonstrate requirements.
 - Workshop: Integrating Python & SQL.
- **Student Deliverables:**
 - **DA 4:**
 - * Assignment Submission: Submit code for feedback.
 - * Discussion Forum: Create a 2-4 minute video showcasing the game and discussing any issues encountered and features added.
 - **Additional Assignment:**
 - * Review Questions for Chapter 14
 - **Bonus Assignment:**
 - * Programming Assignment 8 (Chapter 14 #3)

3.5 Step 5: Final Project

- **Objective:** Combine all the Demonstration Activities into a final game, create a single executable file, and include a ReadMe file.
- **Example Game:** Snake
- **Python Libraries:** os
- **Tools Used:** auto-py-to-exe (GUI for pyinstaller) to create the executable file.
- **Additional Learning:**
 - Resource paths
 - How to get around a known bug in auto-py-to-exe when converting to a single executable file.
- **Resources Provided:**
 - Instructional video using the Snake game to demonstrate requirements.
 - Workshop: Creating ReadMe Files.
- **Student Deliverables:**
 - **Assignment Submission:** Submit final code.
 - **Discussion Forum:**
 - * Create a 2-4 minute video showcasing the game in its entirety, and discussing any issues encountered and features added.
 - * Create a single executable file of the final game.
 - * Include a ReadMe file with instructions on any new game rules and explanations of known quirks in the code.
 - * Play two other classmates' games.
 - * Write a reflective assignment about the course content and structure.
- **Feedback:** Reviews were extremely positive.

4 Evaluation of Course Assessment Criteria

The criteria for the course evaluation worked as desired. Exceptional students achieved scores above 100% due to the bonus assignments, and students who demonstrated a basic understanding of the course received passing grades. No student failed the course with a D or F. All students turned in a final.

5 Lessons Learned and Improvements

5.1 DA 1 Sprint:

- **Observation:** Many students had not made screen recordings with voiceovers before.
- **Improvement:** Created instructional video on how to create professional videos. Will include in future courses.

5.2 DA 2 Sprint:

- **Observation:** Students need to create a project outline to ensure the game is scalable, rather than just seeing the professor's.
- **Improvement:** Required students to submit a project outline after DA 2, and will require this with DA 1 submission in future courses.

5.3 DA 2 Code Review:

- **Observation:** Many students added extra features before completing assignment requirements.
- **Improvement:** Issued reminders to complete requirements before adding extra features. Early creation of personalized project timeline should help resolve this in future courses.

5.4 DA 3 Code Review:

- **Observation:** Students enjoyed GUI programming and found it easy to complete.
- **Improvement:** None needed, but acknowledged students' enthusiasm.

5.5 DA 4 Code Review:

- **Observation:** Students had difficulty integrating SQL with Python.
- **Improvement:** Held an online Coding Workshop about integrating both languages. Will include additional video in future courses.

5.6 Reflective Assignment:

- **Observation:** Students appreciated the lecture videos from previous course as refresher material. Students liked the breakdown of sprints and found the course insightful and skill-building. They valued the detailed feedback given after each DA to improve code. They applauded the bonus assignment opportunities.
- **Improvement:** Continue the approach of sprints but add an additional week to DA 4 deadline.

5.7 Timeline:

- **Observation:** Having other students play game allowed for additional testing and debugging opportunities.

- **Improvement:** Allow students to turn in draft of exe file, so that when they play games and find bugs, this provides feedback so the developer can fix bugs before turning in final submission.

5.8 Workshops:

- **Observation:** Students learned a lot from these workshops.
- **Improvement:** Create videos of these workshop topics for future courses to utilize.

6 Course Materials

6.1 Instructional Videos

- Course Overview & Tips for Success
- Installing & Using Visual Studio Code
- Chapter 3: Decision Structures & Boolean Logic
- Chapter 4: Repetition Structures
- Chapters 5 & 6: Functions, Files & Exceptions
- Chapter 7: Lists & Tuples
- Chapter 8: More About Strings
- Chapter 9: Dictionaries & Sets
- DA 1: Game Origin
- How to Create Professional Videos
- Chapter 10: Classes & Object-Oriented Programming
- Chapter 11: Inheritance
- DA 2: Showing & Keeping Score
- Chapter 12: Recursion
- Chapter 13: GUI Programming
- DA 3: In-game Status Features
- Chapter 14: Database Programming
- DA 4: Leaderboard Created from a Database
- Final: Creating an executable file

6.2 Answer Keys

- Bonus: Programming Assignment 1 Solution
- Chapter 10 Review Questions Answers
- Chapter 11 Review Questions Answers
- Bonus: Programming Assignment 2 Solution
- Bonus: Programming Assignment 3 Solution
- Chapter 12 Review Questions Answers
- Chapter 13 Review Questions Answers
- Bonus: Programming Assignment 4 Solution
- Bonus: Programming Assignment 5 Solution
- Bonus: Programming Assignment 6 Solution
- Bonus: Programming Assignment 7 Solution
- Chapter 14 Review Questions Answers
- Bonus: Programming Assignment 8 Solution

6.3 Workshops

- Python Beginners (Coming from C#)
- Making the Project Scalable
- Integrating Python & SQL
- Writing ReadMe files

6.4 Textbook

- **Title:** Python Programming by Tony Gaddis
- **Chapters Covered:**
 - DA 1: Chapters 3-9
 - DA 2: Chapters 10 & 11
 - DA 3: Chapters 12 & 13
 - DA 4: Chapter 14

6.5 Example Game

- **Snake:** Used to demonstrate requirements for each DA through instructional videos.

7 Project Management

7.1 Planning

- **Project Planning:** Done in the Entry Quiz. Students reviewed the syllabus and answered questions to show understanding of the course outcomes and structure. Scoring 100% on this quiz unlocked the course content.
- **Roadmap Creation:** Initially done by the professor in a general sense, but modified for students to create their own personal roadmaps.
- **Release Planning:** Outlined in the syllabus, which included all the dates and requirements.

7.2 Execution

- **Standups:**
 - **In-Person Classes:** Held twice weekly during class time.
 - **Online Students:** Conducted on an integrity-based system through announcements.
 - **Office Hours:** Students could visit the professor's office three days a week or have video conferences for additional feedback and help.
- **Sprint Reviews:** Conducted through detailed comments on submitted code after each DA.

8 Evaluation and Feedback

- **Review Questions:** Students submit their answers to each chapter through a multiple choice and short answer quiz. Scores were generated automatically, and the short answer answers were reevaluated manually by professor; grades adjusted accordingly.
- **Code Submissions:** Students submit their code at each DA step for feedback and bug fixes.
- **Videos:** Students create 2-4 minute videos for each DA to showcase their work and discuss challenges and features added.

– **Discussion Forum:**

- * **DA 1:** Students helped each other with how to create videos.
 - * **DA 2:** Students helped each other with the scalability of the project.
 - * **DA 3:** Students helped each other with how to fix or add GUI components.
 - * **DA 4:** Students helped each other with SQL integration.
 - * **Final:** Students played each other's games.
- **Check-In:** After DA 2, students provide feedback on their chosen game and discuss its suitability for the course requirements.
 - **Final Project:** The final project includes a single executable file and a ReadMe file, allowing others to play their games, along with a reflective assignment on the course.

9 Conclusion

The ITEC 2270 - Application Development course provides a comprehensive and practical learning experience in modifying games using Python and SQL. Through structured Demonstration Activities and a final project, students gain hands-on experience in coding, debugging, and enhancing their games, while also learning to break a large project into manageable sprints. This approach culminates in positive reviews and reflections on the course content and structure.
