UNIVERSITY OF CALIFORNIA, MERCED

SCHOOL OF ENGINEERING



Object Oriented Programming

CSE 165/ENGR 140

Angelo Kyrilov

5/10/2018

Final Report

Team Polo:

Mitchell Kuiper, Peyton Gylnn

**Project details:** Using a free C++ library called GLUT the team was able to recreate an old game called Breakout using tools learned in class and capitalizing on the power of object-oriented programming. The game was created with the addition of five more classes on top of the given OpenGL program. The main driver function for the project was the Game class which controlled all aspects of the game including its state, score, levels, balls, platform, and logic to combine them. It held pointers to a ball object as well as a platform object to control the game. The ball was the first class implemented in the game and held the logic to bounce a ball around the borders as well as let the speed be controlled via the game. The game could be upgraded with minor changes to allow for the pointer to a ball to be change to a container of Balls rather than just one. The second class implemented was the platform object which was a user-controlled object that received input via the App class which then passed the input to the game which had logic to control the platform. The result from this was a clean app.cpp class which only held the core components needed to drive the game in addition to a timer function to help with the animations of the game. The game class also held a vector called levels, which was a container of all our GameLevel objects which were objects that contained all the information needed to control the current level as well as load a new level into the game. GameLevel accomplished this by holding a vector of objects of the type Bricks which were the blocks at the top of the screen the user is required to destroy to win. One powerful design feature implemented in the project was the games easy ability to add new levels by creating a text document and labeling each brick type with a number. A 0 would be a free space, 1 would be a solid brick, and 2-5 are the different variations of blocks. The code would then build a level based on the given text document and manipulate the size of the blocks based on the dimension of the app as well as how many blocks added to fill up the brick space and load it into the vector of GameLevels. The last class implemented into the game was the Brick object which held a visual representation of a brick using a texture and the logic to deal with when a ball hit it.

**Time Plan/Division of Labor:** The time plan for the project was delayed due to problems and issue we were having with the code. Thankfully, because the team started early the we were able to fix our problems and code in a timely manner to complete the project by the required date. Due to both members being in other upper division CSE classes our time was strict and made meeting up for the project somewhat difficult. We planned to finish the project before finals to give us more time, but the project was ultimately finished later than expected after finals, but we are still happy with the results. This was one of my more successful group project where the team members communicated well. split the labor fairly and completed it when expected to. Peyton and I were able to complete about half the project using pair programming which was also my first successful attempt at and resulted in a clean and efficient program which we both understood and combined the best of our two judgements.

**Lessons Learned:** As stated before, this was my most successful group project, it really helped me learn how to work with teammates to combine thoughts to create a neat and efficient code. This project also helped me learn how to create and design a program before ever writing code, Peyton and I met several times discussing the logistics of the program and designing it on paper before coding. We would then work on the designed code and come back with issue and or improvements to be implemented.