Jacob Stewart

Student # 1174964

COP-2939-8640

Instructor: Dr. Banisakher

Date: 01/15/2024

CAPSTONE PROJECT PROPOSAL

PAC-MAN IN JAVA

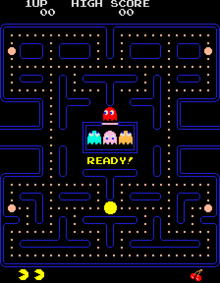


Image from [Pac-Man - Wikipedia](https://en.wikipedia.org/wiki/Pac-Man)

# Introduction

When you stop and think about some of the most early influential video games that started it all a wide slew of Ping-pong-related games come to mind. However, as the capabilities of the GUI started to develop more and more advanced games started to come to life such as Pacman, Galaga, and Super Mario World. These games were highly influential to the world of gaming and to myself personally, as I grew up playing these games in the early 2000s. With how influential video games were in my childhood and with the overall goal to one day become a game developer, hopefully creating more advanced games than Pacman I feel this is an appropriate project to demonstrate my capabilities because of my coursework and independent studying. In conclusion, to demonstrate the capabilities developed throughout my undergraduate career, I intend to make a replica of the original Pacman in full Java.

# Program Writing and Executing

The program will be solely created using Java and its built-in packages namely the AWT (Swing) and the Graphics2D component to render pixels and sprites onto the screen. The program will contain multiple classes that will aim to control game flow, rendering, and control for the player as well as 4 different sets of AI to mimick each ghost's unique states such as scatter and chasing. The program will also include the core concepts of Data Validation and File I/O Implementation. Once completed the program will run through an .jar executable which requires Java to be installed on the target computer. If time allows, I will be attempting to also port the code into a Windows runnable .exe either using a conversion program or attempting to include Java in the resource files for it to access.

This project will use the Java Eclipse environment.

# Required Software and Hardware

The full software requirements to create the program (so far) include:

* Splash screen to require input to start a new game while noting the saved high score from previous runs.
* Moveable Pacman character and ghosts controlled by AI that act on 2 different states (Chase/Scatter)
* Orbs that are collected as the character moves over them and contribute to an overall score and select orbs that cause AI to scatter.
* Resetting the map and applying speed changes to AI movement when all orbs are collected.
* Death interaction and animation for both Pacman and the ghosts.
* Lose and restart screen.
* Functional game loop that accounts for a fixed FPS so draw and update speeds are not accidentally sped up by running on faster computers.

The full Hardware Requirements to run the program (so far) include:

* Windows/Mac/Linux OS with Java installed.
* CPU: i5 processor or newer.
* RAM: minimum 4GB
* SSD/HDD: minimum 1 GB of available disc space.
* Input Device: Keyboard and Mouse

# Project Schedule

I plan to create this project in line with the course assignments to ensure it fully encapsulates all requirements.

The project schedule is defined below:

* Week 1-2 create project proposal and base system and hardware requirements.
* Week 3-4 Refine requirements and create a Test Matric and Use-Case document.
* Week 5-6 Finalize the Test Matrix and Use-Case and start working on the preliminary design.
* Week 7-8 Finalize the preliminary design and begin working on the Detailed Design and mock coding to finalize the Detailed design.
* Week 9-10 Create the Unit Test Suite and create the code skeleton using JUnit.
* Week 10 - April 20th: create code and User manual, as well as presentation for the program.

Throughout the project I imagine I will need to begin implementing features early to create the game in the required timeframe before the code skeleton is completed as with so many moving parts I am sure that the code skeleton will need expanding when it comes down to finally coding and as I am sure bugs in the code start to arise. Outside of this, with the feedback from Dr. Banisakher and others, all my files will be publicly available on my GitHub page, and revisions will be made to all parts of the project as seen fit. At the end of the semester, all files such as Source code, User manuals, and Executables will be uploaded to the canvas assignment Dropbox.

# Risk Analysis

Technical: At first glance, the project I’ve chosen seems relatively simple. However, the complexity of its inner workings is in essence very tedious to implement. Major issues are expected to arise just in the development of it with many of the classes expected to inherit from each other. Visibility and encapsulation also provide room for error as the project grows. Attention to detail is a must.

Schedule: Another issue that needs attention is time management. To appropriately create this project utilizing my time wisely is needed so I am not forced to rush the product leading to even more mistakes and errors that could result in an unusable program.

Operational: Creating this project is expected to take loads of time. I currently am a full-time caretaker for my grandfather and within the last 7 months It’s nearly impossible to sit in one spot for more than 1 hour before needing to attend to him. Outside of this, I work part-time in the evenings 4 days a week. This is the largest risk to the project as I will need to ensure I find the proper time to complete everything required for this course.

Code: I am not entirely well versed in the aspects of pixel rendering nor AI outside of the simple A\* pathing. While I am fully confident that I can implement this I expect my lack of knowledge will ultimately slow down the process in which many errors will be made as debugging will consume plenty of time.

Conclusion: There are a lot of risks included in this project but by ensuring everything is completed in an orderly and efficient fashion through time management and testing the final product should come out fully functional.

# Delivery and Installation

The delivery method of this application will be through Github and potentially free access on Itch.io is allowed. To install and run the user will need to download the code files which will contain a .jar executable to run the game which will need to be stored on the user's hard drive. If allowed the user may also install via the itch.io website receiving a .zip file containing the exported jar file and required files but will not have access to the source code.

# Final Comments

This project’s overall aim is to prove that I can take the core competencies I have developed through my time at HCC and turn it into something that is functional and useable. This project will include a culmination of what I have learned utilizing program flow, data validation, loop control, inheritance and encapsulation, advanced geometrical computing and finally File I/O handling and exception control in an Object-Oriented environment without the use of what I am typically use to when creating games, I.E without the use of Unity and Unreal Engine for rendering and scripting. While I have been through the Intro and Advanced sections of Java at HCC, I consider myself intermediate but no where near advanced as I consider myself in C++ and C however I am looking forward to learning how to render via pixels and expand my knowledge on AI.