

The Life of a Self-taught Software Developer

Software Design Document

Taylor Jensen

Dec 11, 2022

CS 225, Fall 2022

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

INTRODUCTION:

This Software Design Document(SDD) is a simulation of the growth of a self-taught software developer. It takes the player through the journey of how to become a software developer without the conventional method of going to college, and it simulates some of the many situations the player may find themselves in if they were to attempt to go the route of a self-taught developer. The value that this game provides is two-fold: It shows the player some of the obstacles that they may encounter while walking the path of a software developer and gives them the opportunity to see if this is something that they even would like to try at all.

This Simulator game was created to take the users through a simulated journey of rising the ranks in software development and shows some of the hardships of it.

This document contains one main section which is a Problem Description section describing the simulation as implemented.

PROBLEM DESCRIPTION:

This project is a simulation of the journey of a self-taught software developer. This game is meant to be very realistic and is meant to simulate how an ordinary person may go about being a software engineer if they feel conventional learning methods like college or trade school are not their speed. The simulation ends when the player has reached the highest rank of software developer(CEO). It is important to mention that the narrative of this game is in first person so the console will say what the simulated individual is thinking.

The player is prompted to put in their name when the game starts. The simulated individual then says how they need a computer to begin their programming journey, so the player must choose between a Mac, Linux, Windows, or Chrome machine.

Once the player gets home, their programming journey begins! But first, they are instructed to choose which programming language they would like to use between Java, C, Python, and C#. The player then downloads the ide that is best suited for the language they picked. It is now time for the player to choose which means of learning they would like to use. They have the option of utilizing youtube, or a paid online course. The player will then be prompted to write a specific line of code in the language they picked depending on which means of learning they chose. If the user fails, one anxiety point is added to their anxiety counter and they are told to try again but if the anxiety counter reaches 9 points(they start at 0), they lose the game. Once the user successfully writes the specified line of code, they get a job interview.

The player is required to answer a more challenging question in their chosen programming language. Once again, if they get it wrong a point will be added to their anxiety counter. They are prompted to try again until they get it correct. When the player gets the question correct, they officially rank up and are now a Junior Software developer.

The player then gets randomized yearly reports on how they did that year. If they did poorly, they gain an anxiety point. If they do well, they are offered a promotion. If they accept the promotion, they will be required to answer another programming question. Once again if they get it wrong they gain an anxiety point. However, if they get it right, they will be promoted to Senior developer. The same applies to when they are a senior developer, if they get the question right, they will become the CEO and win the game!

PROBLEM SOLUTION:

Computer- This class determines what happens when the player purchases a computer and has four subclasses: Windows, Mac, Chrome, and Linux. This class will print a purchase message when the player chooses a computer and tells the player which specific model was purchased.

Windows- This is a subclass of the Super "Computer". It prints a message when the specific windows computer is opened for the first time. "Congratulations! This computer was a very smart and budget-friendly purchase"

Mac- This is a subclass of the Super "Computer". It prints a message when the specific Mac computer is opened for the first time. "Congratulations! Your bank account hates you!"

Linux- This is a subclass of the Super "Computer". It prints a message when the specific Linux computer is opened for the first time. "Congratulations! You have purchased an overly complex and difficult machine to operate. Good luck, you'll need it."

Chrome- This is a subclass of the Super "Computer". It prints a message when the specific Chrome computer is opened for the first time. "Congratulations! You have purchased a computer that cannot even run a desktop IDE, now you have to use the Web version of the IDE!".

gameController- This class holds the majority of the code and holds the bulk of the functions.

Language- This class stores the player's choice of language that is chosen. For the rest of the game, it will only allow interview/learning questions to be asked about the specified language until the game ends. This class has attributes name, questions, answers, and ide which determine which specific questions to ask and which IDE to install depending on the language that the player picks.

Programmer- This is the most in-depth of the classes in this project. This class has a name, rank, computer, language, and anxiety, and paidCourse attribute. The name attribute stores what the player wants to call themselves. Rank stores what rank the player is depending on the player's

progress in the game. Computer stores which computer the player has. Language stores which language the player have chosen. This occurs until they get a job. Anxiety stores what anxiety level the player is at from 0 to 9. This class is in charge of increasing or decreasing the player's anxiety level and keeps track of all of the values that the player gives them.

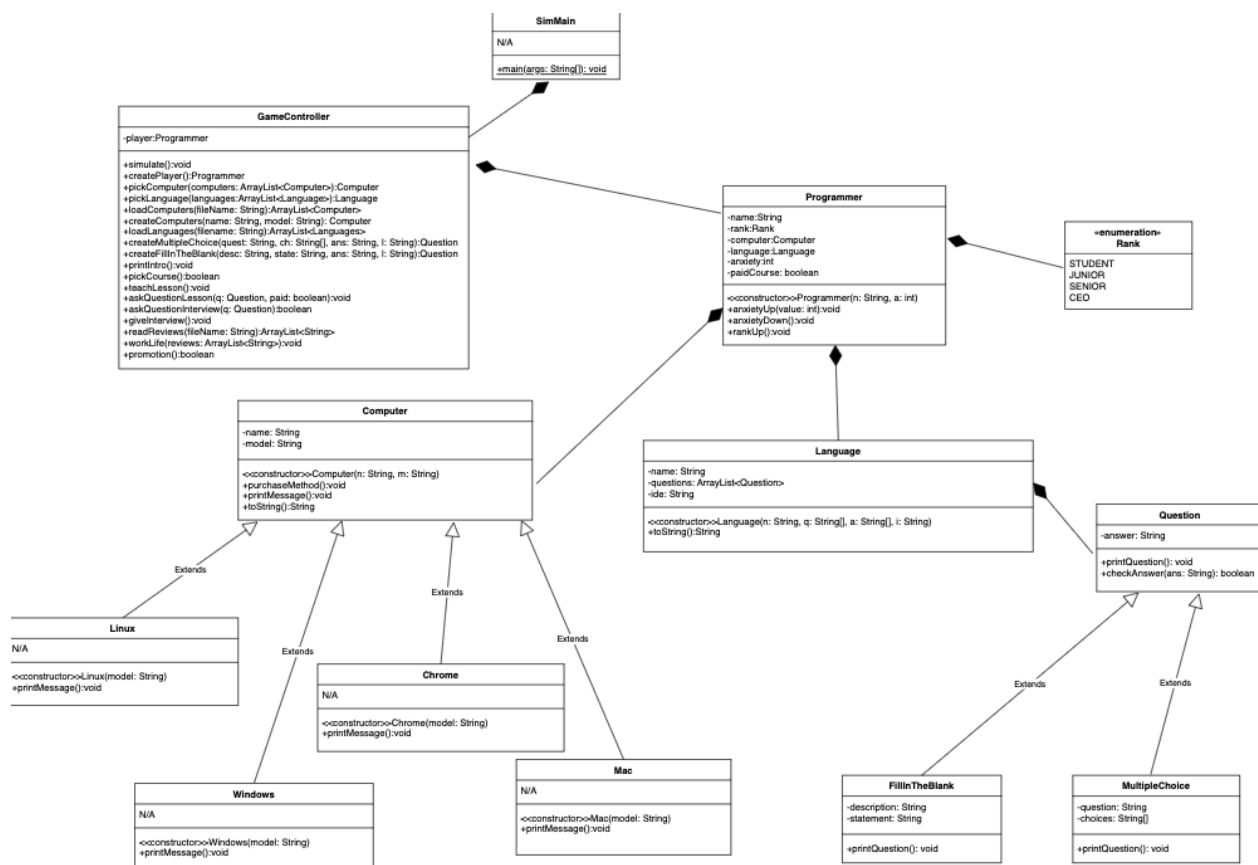
SimMain- This class runs the program. It is the main class of the program and its only purpose is to start and end the game.

Question- This class is the superclass of FillInTheBlank and MultipleChoice and stores the question types.

FillInTheBlank- This class classifies all questions that are fill in the blank and sorts the fill in the blank questions from the multiple choice questions.

MultipleChoice-This class classifies all questions that are multiple choice and sorts the multiple choice questions from the fill in the blank questions.

Rank- This is an enum class that simply holds the types of rank.



REFERENCES:

APPENDICES:

This is optional but may include external sources, source code, or other related material.

[Shall be completed as needed with each deliverable.]