

Project 3 Requirements Document

This document outlines previous project work and requirements for Project 3. It includes summaries of past programming assignments and the current feature and sprint requirements for the new project.

Previous Projects

Ref#	Story No.	Story Pool	Story Source	Summary (25–50 words)
1	1	EECS 168 Board Work	Python	Printed 'Hello World' to the terminal as a first introduction to Python
2	1	EECS 168 Lab 2	Python	Calculated expected number of people sick with the flu using a probability
3	1	EECS 168 Lab 3	Python	Created a guessing game similar to Wordle, tracking guesses and
4	1	EECS 168 Lab 9	Python	Developed a Circle class with attributes like diameter, area, and c
5	2	EECS 581 Assignment 1	Python	Built an easy-difficulty Minesweeper AI that selects squares random
6	2	EECS 210 Assignment 1	Any Language	Program checked truth statements with P, x, and y to determine tr
7	2	EECS 468 Lab	Haskell	Created 'Hello World' program in five different ways to explore out
8	3	EECS 268 Assignment 1	C++	Simulated a blob that moves through a map and eats people; inclu
9	3	EECS 581 Use Case 1	Text Analysis	Analyzed project use cases for complexity using keywords and ar
10	3	EECS 210 Assignment 1	Python	Solved Sudoku puzzles by filling zeros and outputting completed t
11	3	EECS 268 Lab 1	C++	Read and sorted data from a file by rating or sales, then displaye
12	3	EECS 268 Lab 6	Python	Implemented a Binary Search Tree with preorder, inorder, and pos
13	5	EECS 268 Lab 7	Python	Built a Pokédex using file I/O and BST for searching Pokémon dat
14	5	EECS 348 Lab 6	C++	Performed matrix addition and subtraction based on user input.
15	5	EECS 348 Final Project	C++	Parsed and tokenized algebraic expressions to simplify logical op
16	8	EECS 565 Assignment 1	Linux VM	Simulated a buffer overflow attack in a controlled, randomized-fre
17	8	EECS 388 Final Project	C++	Created a simulated self-driving car responding to LIDAR and can
18	8	EECS 678 Buddy Alloc	Project	Simulated memory allocation and recovery using a buddy system
19	8	EECS 330 Directed Graph	Graph	Developed a directed graph structure using previous data structur
20	13	EECS 510 Final Project	Custom Language	Created a formal language for FSM/PDA/Turing Machine and sim
21	13	EECS 468 Lab 8	Haskell	Parsed arithmetic instructions and tokenized input for computatio
22	13	EECS 678 Quash Project	Project	Simulated a shell that executed terminal commands like ls, rm, an

Project 3 Requirements

ID	Description	Story Points	Priority	Sprint No.	Notes
1	Get Godot setup and installed	2	1	1	Install the Godot game engine
2	Familiarize with Godot	3	1	1	Take tutorials and explore interface
3	Create spaceship movement	5	1	1	Implement acceleration and realistic movement
4	Create environment	3	1	1	Add asteroids and objects
5	Collision detection between asteroid and rocket	2	1	1	Detect collisions; player dies on hit
7	Deliver packages between two points	3	1	2	Enable transport objectives
8	Create space station/base	2	1	2	Add respawn and upgrade base
9	Create golden asteroid collectible	2	2	2	Add resource-based asteroids
10	Mini map	5	2	3	Show player surroundings
11	Fuel depletion	2	2	3	Fuel reduces with movement; affects speed
12	Ship upgrades	1	2	3	Allow ship improvement (fuel, speed, etc.)
13	Resource management	1	2	3	Handle variables tied to resources
14	Quest manager/log	5	2	Final	Track and manage tasks
15	Oxygen depletion	1	3	Final	Player dies if oxygen runs out
16	Testing and debugging	8	1	Final	Fix game issues and ensure stability
17	Game balance	2	1	Final	Ensure fair and fun gameplay experience