

Module #4 Plan | CSE 310 – Applied Programming

Name	Date	Teacher
Nathan Luckock		Bro McGary

Software Description

A classic Snake game built with JavaScript that includes both manual play and an AI mode. The game will have multiple difficulty levels, score tracking, and a simple AI algorithm that automatically plays the game. Players can switch between manual control and watching the AI play, with features like game speed adjustment and collision detection.

Module

Mark an **X** next to the module you are planning

Module	Language	
Cloud Databases	Java	
Data Analysis	Kotlin	
Game Framework	x	R
GIS Mapping	Erlang	
Mobile App	JavaScript	x
Networking	C#	
Web Apps	TypeScript	
Language – C++	Rust	
SQL Relational Databases	Choose Your Own Adventure	

Create a Schedule

Create a detailed schedule using the table below to complete your selected module during this Sprint. Include details such as what (task), when (time), where (location), and duration. You should also include time to work on your team project. You are expected to spend 16 hours every Sprint working on your individual module, team project, and other activities. Time spent on this individual module should be at least 10 hours.

	First Week	Second Week
Monday	Set up project, basic game board and snake movement (2 hours)	Add difficulty levels and score tracking (1 hour)
Tuesday	Implement food generation and collision detection (2 hours)	Polish UI and add game over screen (1 hour)

	First Week	Second Week
Wednesday	Add game controls and basic AI algorithm (2 hours)	Testing and bug fixes (1 hour)
Thursday	Improve AI logic and game speed controls (1 hour)	Create video demo and documentation (2 hours)
Friday	Add game restart and pause features (1 hour)	Final UI polish and features (1 hour)
Saturday	Write README and documentation (1 hour)	Final testing and GitHub push (1 hour)

Identify Risks

Identify at least two risks that you feel will make it difficult to succeed in this module. Identify an action plan to overcome each of these risks.

	Risk	Action Plan
1	AI algorithm might be too complex to implement	Start with a simple pathfinding algorithm (like always moving toward food) and iterate
2	Game physics and collision detection bugs	Test collision detection early and often, use console.log for debugging