SE 4450 Software Engineering Design II
Team 32 *Capstone Project*

STELLA

Simulated Training Environments and Large Learning Automata

Sprint Plans 1 & 2

Members	Beaudry Hajji, Isabelle	ibeaudr@uwo.ca
	Hoffman, Madelyn	mhoffm32@uwo.ca
	Bazina-Grolinger, Dominic	dbazinag@uwo.ca
	Su, James	tsu32@uwo.ca
Advisors	Yimin Yang	yimin.yang@uwo.ca

November 6, 2024

Western University

Contents

1 | Team Details

1.1 | Team Roles

Team Member	Roles
James Su	Scrum Master Developer
Dominic Bazina-Grolinger	Developer
Isabelle Beaudry Hajji	Developer
Madelyn Hoffman	Developer

1.2 | Meeting Schedule

The team will meet 1-2 times per week. Meetings will be 2 hours in length with scheduled sessions as follows:

- Wednesdays: 3:30 PM - 5:30 PM when class not in session
- Fridays: 11:30 AM - 1:30 PM

These regular meetings will ensure ongoing progress tracking, collaborative problem-solving, and alignment on project milestones. Additional sessions may be scheduled as needed to address any specific challenges or issues that may arise, or to review project phases.

2 | User Stories and Point Estimates

2.1 | User Stories

Listed below are STELLA's project requirements in the User Story format. The primary user of STELLA will be other developers and system integrators looking to implement our agent and tools into new techniques or physical models. We want to ensure that STELLA's toolkit is efficient to use, while easy to monitor and modify during development.

ID	Name	User Story
		Configuration
1	Configuration File	As a system integrator, I would like to tune the agent's hyperparameters so that the model can be adjusted to learn properly (Without GUI options).
2	GUI Runtime and Configuration	As a system integrator, I want to be able to conveniently edit and adjust the hyperparameters and start training from a GUI, so that it is easy to train new employees and maintainers.
3	Cost Function, Optimizer, and RL Algorithm Selection	As a system integrator, I want the option to select and vary preferences algorithm variations so that I can efficiently refine the agent's training.
4	Checkpointing and Transfer Learning	As a system integrator, I want to save agent's progress to incrementally improve its performance by retaining insights from past simulations, for steady performance and growth.
		CARLA Interaction
5	Basic Movement	As a system integrator, I want the agent to be able to move and interact with the CARLA simulator so that it can learn from it and be evaluated.

ID	Name	User Story
6	Sensor Integration for Environment Feedback	As a system integrator, I want the agent to be able to receive sensor information (e.g., cameras, LiDAR, radar) so that the agent can receive feedback on its actions.
7	Sensor Data Preprocessing	As a system integrator, I want the agent to be able to properly perceive and utilise sensor information so that the agent can make the correct decisions.
	Logging and Monitoring	
8	Simulation Metrics Monitoring	As a system integrator, I want to track and evaluate the agent's performance in specific tasks so that the agents abilities in different areas can be identified for optimization.
9	Training Metrics Monitoring	As a system integrator, I want detailed logging of the agent's training metrics, so I can identify issues in training. (Differs from simulation metrics as the data source, processing, and destination are entirely different)
10	Integration with External Monitoring Tools	As a system integrator, I want the ability to integrate CARLA's simulation output with external monitoring tools (Weights and Biases) for advanced analysis and visualization of performance metrics and trends
11	Detailed Error Reporting	As a system integrator, I want detailed error reporting that includes error type, context, and relevant metrics so I can quickly diagnose and resolve issues affecting system performance and stability.
	Basic Driving Manoeuvres	
12	Regular Stopping	As a system integrator, I want the agent to be able to stop when needed at the right position, like when a stop sign is detected (does not

ID	Name	User Story
		include detection of stopping conditions) so that it can follow traffic rules correctly.
13	Turning	As a system integrator, I want the agent to be able to turn at intersections when needed so that it can follow traffic rules correctly.
14	Adaptive Speed Management	As a system integrator, I want the agent to adjust its speed based on road conditions, traffic flow, and legal speed limits, so that it drives safely and efficiently in various environments.
15	Stationary Obstacle Avoidance	As a system integrator, I want the agent to detect and avoid obstacles (ex: traffic cones, parked cars, etc.) so that safety can be ensured in real-world challenges.
	Complex Manoeuvres	
16	Lane Following	As a system integrator, I want the agent to be able to follow the curve of the road in so that it won't drive off the road or drift into other lanes.
17	Lane Changing	As a system integrator, I want the agent to perform safe lane changes in order to avoid collisions so that I can assess its awareness and responsiveness on the road
18	Moving Obstacle Response	As a system integrator, I want the agent to respond efficiently to abrupt interruptions on the road (ex: unexpected pedestrian, large animals, etc.) so that its reliability in critical situations can be ensured.
19	Pedestrian Interaction	As a system integrator, I want the agent to react appropriately to pedestrians, including yielding and adjusting speed, so I can validate its safety around human interactions.

ID	Name	User Story
20	Traffic Following	As a system integrator, I want the agent to be able to work in different traffic conditions (ex: speed of traffic, number of cars, number of pedestrians, etc) to ensure the performance of the reinforcement learning model does not change due to traffic.
21	Road Sign Recognition	As a system integrator, I want the agent to recognize and respond to road signs (e.g., speed limits, stop signs) to ensure it can obey traffic laws.
	Environment Diversity	
22	Location Agnostic Driving	As a system integrator, I want the agent to adapt its driving behavior to different location settings (e.g., rural roads, city streets, highways) so I can confirm its versatility and reliability across diverse driving environments.
23	Environmental Conditions	As a system integrator, I want the agent to adjust its driving behavior according to different times of the day (e.g., broad day, dusk, night), and weather to ensure it performs reliably under varying environmental conditions.
	Non-Functional Requirements	
24	Modularity and Code Cleanliness	As a future contributor, I want components of the agent and model to be modular and well documented, so I can easily update or modify the STELLA tookit.
25	Robustness	As a system integrator, I want the agents behavior to be robust against edge cases and adversarial situations.
26	Efficiency	As a system integrator, I want the toolkit to run efficiently so that it makes the best use of my resources.

2.2 | Poker Game Estimate

To determine the number of story points for each requirement, we used the Poker Game Estimate. Story points are limited to values in the Fibonnaci Sequence (1, 2, 3, 5, 8, ...) to describe the order of magnitude of effort required for each story. The poker game estimate involves the following process:

1. Each team member individually picks a value for the story points.
2. The group shares their decisions and rationale with each other.
3. If there is a consensus, the process is complete.
   If not, each member reconsiders their valuation and the process repeats

Below, the final number of points for each user story is shown, as well as the reasoning for each value that the group agreed upon.

ID	Name	Points	Conclusive Reason
1	Configuration File	1	This story only requires creating an interface for adjusting hyperparameters through a configuration file.
2	GUI Runtime and Configuration	2	The GUI is UI-focused with limited backend complexity. It provides essential features for ease of use but requires minimal integration of new technology.
3	Cost Function, Optimizer, and RL Algorithm Selection	3	Adjusting RL reward functions is minimally complex but will likely have to be modified for each simulation task (Weather changes, Traffic, emergency situations) to find the optimal algorithm for each.
4	Checkpointing and Transfer Learning	8	Checkpointing for future use will be fairly time consuming as it requires creating and optimizing functions for saving the agent's progress at specified intervals as well as functions for using insights from past simulations.
5	Basic Movement	8	This story was given 8 points due to its critical importance as the foundation of

ID	Name	Points	Conclusive Reason
			our entire project. The strength and accuracy of our base environment are key to successfully implementing future user stories that will build upon it, so it will require additional testing and refinement.
6	Sensor Integration for Environment Feedback	3	This user story will take a moderate amount of time, as CARLA allows simple and easy generation and handling of sensor data; however, functions for forwarding information to the agent will need to be implemented.
7	Sensor Data Preprocessing	4	Data preprocessing can be challenging and must match the format the model was pretrained on.
8	Simulation Metrics Monitoring	3	Straightforward setup for logging performance metrics with CARLA's output, but is essential for tracking progressive improvements.
9	Training Metrics Monitoring	3	Detailed logging is fairly straightforward but critical for monitoring training.
10	Integration with External Monitoring Tools	5	Moderate complexity to set up data streaming with external tools like WandB, essential for visualizing trends and analysis during testing.
11	Detailed Error Reporting	5	Effective error logging is essential for iterative development and testing; moderately complex, as it demands comprehensive logging with contextual data.
12	Regular Stopping	5	Stopping will depend on speed control, perception and predefined legal constraints; moderately complex with

ID	Name	Points	Conclusive Reason
			some integration of dynamic traffic flow considerations.
13	Turning	5	Turning relies on speed control as well as perception and predefined legal constraints; moderately complex with some integration of dynamic traffic flow considerations.
14	Adaptive Speed Management	5	Speed control relies on perception and predefined legal constraints; moderately complex with some integration of dynamic traffic flow considerations.
15	Stationary Obstacle Avoidance	8	This basic maneuver is given 8 story points due to safety requirements and the challenging sensor fusion and real-time decision-making needed to handle diverse road elements.
16	Lane Following	5	While lane following is a well covered topic in previous methods, it can still be difficult to implement and requires some research and understanding.
17	Lane Changing	5	Moderately complex due to necessary responsiveness, real-time adjustments, and safety requirements but manageable with existing perception and control components.
18	Moving Obstacle Response	8	Requires complex algorithms for handling sudden, varied events (e.g., sudden stops), needing in-depth testing and adaptability.
19	Pedestrian Interaction	8	Interaction with pedestrians adds complexity due to unpredictable behavior. Extra testing and edge-case consideration

ID	Name	Points	Conclusive Reason
			necessary for thorough safety validation in human-centric scenarios.
20	Traffic Following	8	Adapting to varied traffic aligns with CARLA's capabilities but requires complex perception and real-time response adjustments to maintain safety.
21	Road Sign Recognition	8	Recognizing road signs involves image recognition and classification, requiring extensive model training and testing.
22	Location Agnostic Driving	8	This user story will be moderatley complex as it will require implementing logic to detect the current driving environment as well as creating behaviour models for each driving evironment.
23	Environmental Conditions	8	Adapting to weather adds substantial testing requirements, given the project's progressive complexity and real-time adjustment needs.
24	Modularity and Code Cleanliness	3	Maintaining clean code and component documentation will take slightly more time throughout development but will ease other other areas of development.
25	Robustness	8	Robustness requires careful hand selection of edge-case scenarios for testing, and large datasets for training.
26	Efficiency	8	Constant, difficult effort that will require understanding of the underlying mechnisms like CUDA, Operating Systems, and Neural Networks.

3 | Sprint One

3.1 | Sprint Overview

The goal of this sprint is to have a very simple and working agent. It will not be refined or optimized fully, but we would like to have a good foundation for testing and optimization for the next sprint. The goal is to have a working CARLA environment, as well as an agent that can interact with that environment. Performance metrics are not important, but we want to be able to view those metrics for future use. We start by configuring the simulation environment, and build a development framework around it. Pipelines must be setup to transfer information back and forth efficiently. Then, an agent can be started. This initial agent only needs to set the foundation for later work, like being able to receive images from the simulator, and send actions in return. With a preliminary agent completed, metrics and performance data needs to be collected and sent to loggers so that they can be used when testing the agent. Upon completion, we have a fully functional environment that can take autonomous driving algorithms in a modular way and test them in CARLA.

3.2 | Risks and Challenges

This sprint is important for later work, as it sets the foundation for sprint 2. If the structure of the initial agent is inadequate or inflexible, it could limit further testing and optimization. Setting up and integrating the CARLA simulator for basic movement and sensor initialization is a crucial step in providing a solid foundation for our agent. If any issues arise in configuring CARLA or integrating its API, this could delay all other tasks as CARLA is the primary environment for the agent's operations. Initializing and integrating camera and LiDAR sensors, as well as preprocessing their data, may pose a risk as sensor data could be too complex or noisy. This may complicate the process of converting it into usable formats for the agent, delaying sensor integration and impacting real-time testing and performance tracking. Ensuring the agent can effectively interpret sensor data without compromising on performance is a challenge, especially if responsiveness is affected by large data volume. Aside from technical risks and challenges, effective communication and collaboration will be the most essential to avoid delays. Misalignment on priorities or insufficient sharing of updates may impact the overall progress and quality of the sprint deliverables.

3.3 | Team Working Agreement

Team Working Agreement	
Work Scheduling	
Working Hours	Monday: 4pm to 8pm
	Tuesday 4pm to 8pm
	Wednesday 6pm to 9pm
	Thursday 4pm to 8pm
Daily Scrum	12 PM - 12:15 PM
Collaboration	
Communication and Tools	Discord
Project Planning	GitHub Projects
Collaboration and Version Control	GitHub
Goals	
Sprint Goal	To have a simple agent that will provide a good foundation for testing and optimization in the next sprint.

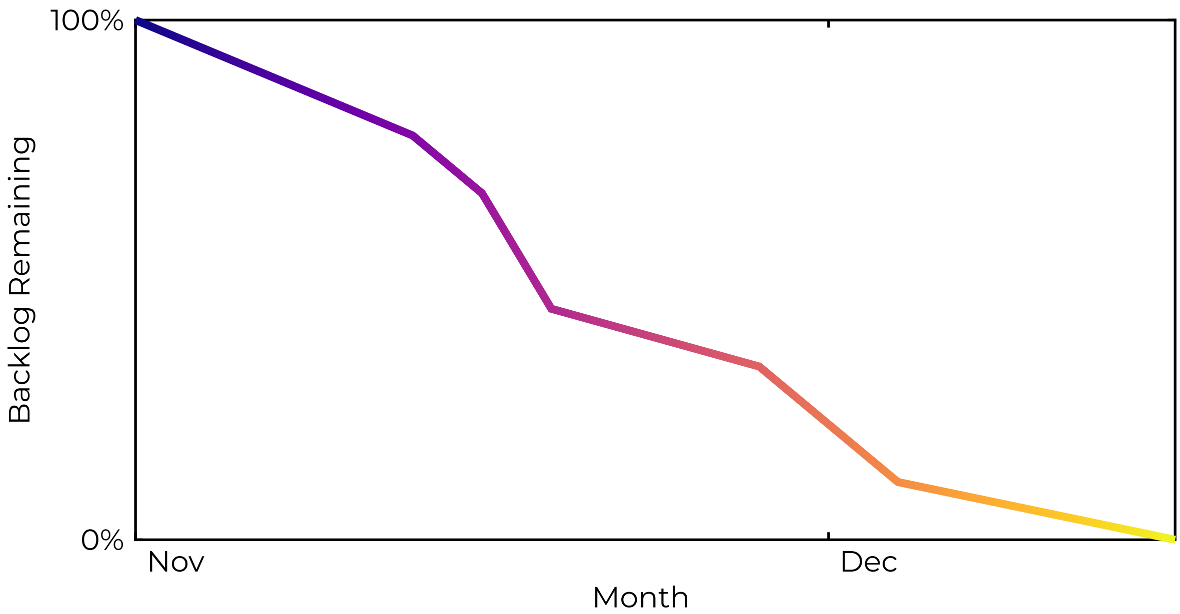
3.4 | User Stories and Tasks

User Story	Task ID	Task Description	Workload Estimation	Assignee/ Due Date
1. Configuration File	1.1	Write a configuration parser for training hyperparameters as well as a specification for config files	2 hours	J.S Nov 18
2. GUI Runtime and Configuration	2.1	Develop a simple, user-friendly tool for adjusting key	2 hours	D.B Dec 15

User Story	Task ID	Task Description	Workload Estimation	Assignee/ Due Date
		parameters (ex: speed, reaction time, etc.).		
	2.2	Use the configurator generate a usable config file	30 mins	I.B.H Dec 15
5. Basic Movement	5.1	Set up the CARLA environment in a way that is accessible to the development team.	5 hours	M.H Nov 5
	5.2	Integrate CARLA's API to enable basic movement commands (ex: start, stop, turn).	3 hours	M.H Nov 5
	5.3	Run test cases to confirm the agent can move forward, stop, and turn, documenting any issues for later refinement.	3 hours	I.B.H Nov 5
6. Sensor Integration for Environment Feedback	6.1	Initialize camera and LiDAR sensors in CARLA and integrate them into the agent.	4 hours	D.B Nov 11
7. Sensor Data Preprocessing	7.1	Develop data parsers to preprocess sensor data and convert raw data into a format that the agent may use.	3 hours	J.S Nov 15
9. Training Metrics Monitoring	9.1	Develop a logging module to initialize and organize logging	2 hours	I.B.H Nov 18

User Story	Task ID	Task Description	Workload Estimation	Assignee/ Due Date
		categories (ex: errors, warning, metrics) and save logs in a structured format (ex: JSON, CVS) to make analyzing more simple.		
	9.2	Track and log important metrics such as loss, reward progression, training completion, and agent performance metrics in real-time.	6 hours	D.B Nov 18
10. Integration with External Monitoring Tools	10.1	Implement the function to stream CARLA's output data to WandB.ai in real-time, allowing metrics to be visualized at the simulation runs.	1 hour	M.H Feb 15
24. Modularity and Code Cleanliness	24.1	Establish a high-level structure that separates the agent's core modules (ex: perception, control, reward function, environment interaction).	1 hour	M.H Nov 26
25. Robustness	25.1	Test for robustness and refine dataset.	4 hours	D.B Dec 3

3.5 | Burn Down Chart



4 | Sprint Two

4.1 | Sprint Overview

The goal of this sprint is to utilize the environment built in the first half. A well built environment should allow for rapid testing and prototyping of agent variants, which we will combine and test in a variety of ways. The development environment will also be improved continually, through the addition of features like checkpointing. Testing and optimization will also be an important part of this sprint, to allow for the best performance in the final deliverables.

4.2 | Risks and Challenges

The complexity of the framework quickly builds throughout sprint 2. The challenge of integrating and testing complex new features could lead to unexpected behaviour or compatibility issues within the agent. Ensuring compatibility across different components could lead to dependency issues that may affect the overall system stability. Real-time data processing for tasks may also pose as a challenge to the system's performance. Tasks like error logging, metric monitoring, and environmental condition adjustments could put strain on the system under complex scenarios or heavy load. Given the wide range of tasks and features targeted for completion in this sprint, meeting deadlines could be a challenge. Limited time may affect the ability to test and refine complex functionalities. Time management, consistent meetings, and thorough communication will be key to meeting deadlines and delivering a completed and optimized final product.

4.3 | Team Working Agreement

Team Working Agreement	
Work Scheduling	
Working Hours	TBD (schedules pending)
Daily Scrum	12 PM - 12:15 PM
Collaboration	
Communication and Tools	Discord
Project Planning	Github Projects

Team Working Agreement	
Collaboration and Version Control	GitHub
Goals	
Sprint Goal	Polish sprint 1 agent and add more complicated features.

4.4 | User Stories and Tasks

User Story	Task ID	Task Description	Workload Estimation	Assignee / Due Date
3. Cost Function, Optimizer, and RL Algorithm Selection	3.1	Create a dropdown or selection tool to choose between RL algorithms.	2 hours	I.B.H Feb 4
	3.2	Allow customization of optimizer parameters such as the learning rate and momentum.	1 hour	J.S Feb 4
4. Checkpointing and Transfer Learning	4.1	Implement a function to save the agent's progress at specified intervals.	4 hours	J.S Jan 26
	4.2	Implement a function to use insights from past simulations to improve the agent's performance and growth.	2 hours	I.B.H Jan 26
8. Simulation Metrics Monitoring	8.1	Set up a logging system to record key performance metrics (ex: reaction time,	2 hours	M.H Jan 7

User Story	Task ID	Task Description	Workload Estimation	Assignee / Due Date
		obstacle avoidance success)		
11. Detailed Error Reporting	11.1	Identify and categorize common error types (ex: sensor failure) and specify the context data to capture with each error type.	1 hour	M.H Jan 21
	11.2	Develop a centralized logging system that captures error type, time, and context data to then output to a log file.	2 hours	I.B.H Jan 21
12. Regular Stopping	12.1	Implement logic for detecting appropriate stopping points such as stop lines.	3 hours	D.B Feb 12
	12.2	Implement stop confirmation which will ensure the agent halts at the correct position and verifies stop completion.	3 hours	J.S Feb 12
13. Turning	13.1	Implement turn detection for when a turn is needed based on map data or visual cues.	3 hours	J.S Jan 26
	13.2	Program the agent to initiate and complete	3.5 hours	D.B Jan 26

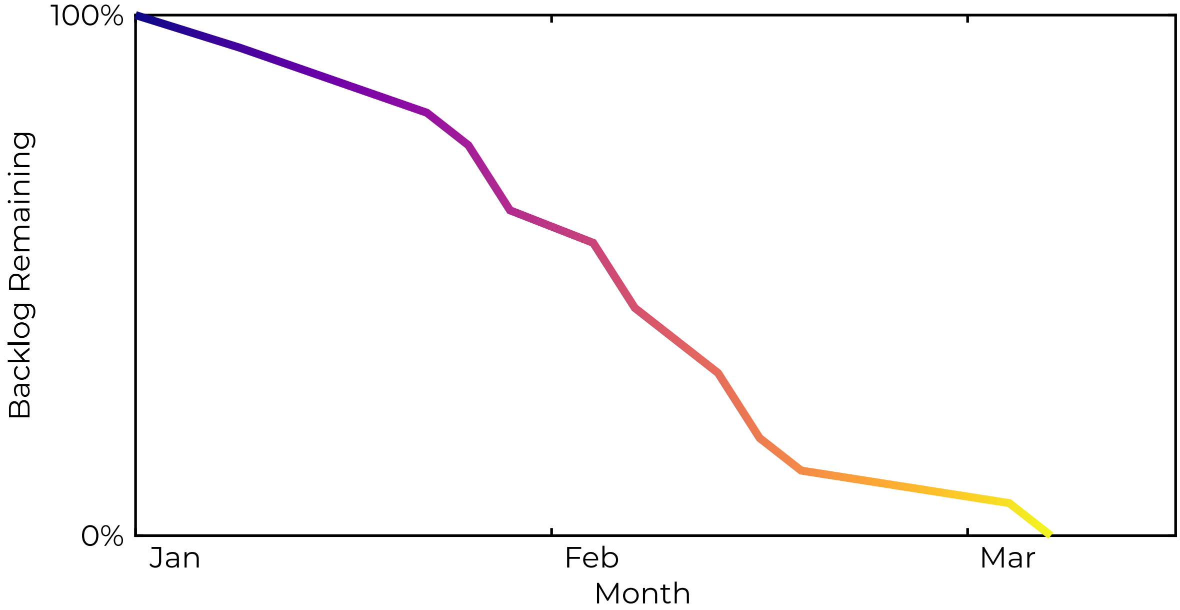
User Story	Task ID	Task Description	Workload Estimation	Assignee / Due Date
		turns within lane boundaries.		
14. Adaptive Speed Management	14.1	Implement an algorithm that adjusts speed based on road conditions, traffic flow, legal speed limits, etc.	5 hours	M.H Feb 17
	14.2	Add logic to respond dynamically to traffic flow, such as slowing down in sudden heavy traffic.	3 hours	M.H Feb 17
15. Stationary Obstacle Avoidance	15.1	Use sensor inputs to detect stationary obstacles in the path.	1 hour	D.B Feb 2
	15.2	Implement pathfinding logic to navigate around obstacles	2 hours	I.B.H Feb 2
	15.3	Set up alerts or actions if the agent comes too close to a detected stationary obstacle.	1 hour	J.S Feb 2
16. Lane Following	16.1	Develop a lane detection algorithm that tracks lane markngs and road edges to allow the agent to remain within the boundaries of the road.	3 hours	J.S Feb 12

User Story	Task ID	Task Description	Workload Estimation	Assignee / Due Date
	16.2	Code agents control parameters to handle road curves smoothly in order to prevent lane drift	5 hours	I.B.H Feb 12
17. Lane Changing	17.1	Use sensor data / camera inputs to detect available lanes and check for nearby vehicles / obstacles in ajacent lanes.	1 hour	D.B Feb 15
	17.2	Develop control algorithm to smoothly and safely maneuver the agent from the current lane to the target lane, while maintaining the proper speed and orientation.	5 hours	M.H Feb 15
18. Moving Obstacle Response	18.1	Allow agent to predict movement patterns of obstacles (ex: pedestrian crossing prediction)	3 hours	I.B.H Feb 6
	18.2	Implement a functionality to adjust paths dynamically based on a moving obstacle's trajectory.	4 hours	D.B Feb 6
19. Pedestrian Interaction	19.1	Implement pedestrian detection using sensors / camera data	2 hours	J.S Feb 15

User Story	Task ID	Task Description	Workload Estimation	Assignee / Due Date
		and track movement to predict potential paths.		
	19.2	Develop logic for the agent to slow down or stop when pedestrians are detected within a certain range, including yielding as necessary.	3 hours	J.S Feb 15
20. Traffic Following	20.1	Develop a module to assess traffic density and flow, adapting the agent's behaviour based on the number of nearby vehicles / pedestrians.	4 hours	D.B Jan 23
	20.2	Enable agent to adjust its speed base on speed of nearby vehicles to ensure safe navigation.	3 hours	I.B.H Jan 23
21. Road Sign Recognition	21.1	Integrate an image recognition model specifically for detecting road signs.	5 hours	M.H Jan 21
	21.2	Add logic to interpret signs and adjust agents behaviour based on the sign (ex: stopping for stop sign,	5 hours	D.B Jan 21

User Story	Task ID	Task Description	Workload Estimation	Assignee / Due Date
		adjust speed for speed limits).		
22. Location Agnostic Driving	22.1	Create behaviour modules tailored for specific locations that adjust the agent's driving style based on road type and expected traffic.	4 hours	I.B.H Mar 2
	22.2	Implement the logic to detect the current driving environment and switch to the corresponding behaviour module.	4 hours	J.S Mar 2
23. Environmental Conditions	23.1	Add mechanisms to adjust the agent's perception and behaviour based on the time of day (ex: use headlines at dusk).	4 hours	J.S Mar 6
	23.2	Integrate adjustments for different weather conditions (ex: lower speeds in rain, snow, fog, etc).	5 hours	I.B.H Mar 6

4.5 | Burn Down Chart



5 | Project and Sprint

5.1 | Using the Backlog

The purpose of the project backlog is to help track the group's project as we complete each story task. Currently, a few user stories are already underway, as each story is completed it will be marked off, but it must fit our acceptance criteria as outlined later on in this section. In the case that problems do arise and story points cannot be completed in their allotted time frame a prioritization discussion will commence at the next stand-up meeting. During this time the priority and importance of that story will be determined; if necessary, timelines for story completions may be adjusted. The backlog also contains user stories that are not covered in either sprint, but may be implemented continuously or if time allows.

Stories marked as "Complete" must be:

- Reviewed by at least one other group member
- Fully functional and integrated into the project
- Every class/function/etc documented with relevant instructions or comments
- Tested thoroughly by hand or automated test cases depending on the specific context

User Story / Task	Not Started	In Progress	Complete
1. Configuration File			
1.1 Write parser		✓	
2. GUI Runtime and Configuration			
2.1 Build GUI	✓		
2.2 Config Generator		✓	
3. Cost Function, Optimizer, and RL Algorithm Selection			
3.1 Create RL algorithm dropdown	✓		
3.2 Customization of optimizer parameters	✓		
4. Checkpointing and Transfer Learning			

User Story / Task	Not Started	In Progress	Complete
4.1 Function to save progress	✓		
4.2 Function to use past insights	✓		
5. Basic Movement			
5.1 Environment setup		✓	
5.2 Movement pipeline	✓		
5.3 Movement testing	✓		
6. Sensor Integration for Environment Feedback			
6.1 Initialize camera and LiDAR sensors	✓		
7. Sensor Data Preprocessing			
7.1 Develop data parsers	✓		
8. Simulation Metrics Monitoring			
8.1 Set up logging system	✓		
9. Training Metrics Monitoring			
9.1 Develop logging module	✓		
9.2 Track / log metrics	✓		
10. Integration with External Monitoring Tools			
10.1 Write data streaming code		✓	
11. Detailed Error Reporting			
11.1 Categorize common error types	✓		
11.2 Develop logging system	✓		
12. Regular Stopping			
12.1 Implement stop detection	✓		
12.2 Implement stop confirmation	✓		
13. Turning			

User Story / Task	Not Started	In Progress	Complete
13.1 Implement turn detection	✓		
13.2 Program turning action	✓		
14. Adaptive Speed Management			
14.1 Implement speed adjustment algorithm	✓		
14.2 Dynamic traffic response logic	✓		
15. Stationary Obstacle Avoidance			
15.1 Detect obstacles with sensor inputs	✓		
15.2 Implement pathfinding logic to navigate around obstacles	✓		
15.3 Alerts for obstacles in close vicinity	✓		
16. Lane Following			
16.1 Lane detection algorithm	✓		
16.2 Lane detection algorithm	✓		
17. Lane Changing			
17.1 Use sensors to detect nearby lanes	✓		
17.2 Develop control algorithm	✓		
18. Moving Obstacle Response			
18.1 Movement pattern prediction	✓		
18.2 Function to adjust paths dynamically	✓		
19. Pedestrian Interaction			
19.1 Implement pedestrian detection	✓		
19.2 Logic to slow down when pedestrian nearby	✓		

User Story / Task	Not Started	In Progress	Complete
20. Traffic Following			
20.1 Develop traffic assessment module	✓		
20.2 Enable natural speed adjustment	✓		
21. Road Sign Recognition			
21.1 Integrate an image recognition model specifically for detecting road signs.	✓		
21.2 Add logic to interpret signs and adjust agents behaviour based on the sign (ex: stopping for stop sign, adjust speed for speed limits).	✓		
22. Location Agnostic Driving			
22.1 Create behaviour modules	✓		
22.2 Logic to detect driving environment	✓		
23. Environmental Conditions			
23.1 Adjust behaviour based on time-of-day	✓		
23.2 Weather condition adaptation	✓		
24. Modularity and Code Cleanliness			
24.1 Design modular structure		✓	
25. Robustness			
25.1 Improve dataset	✓		
26. Efficiency			
26.1 Optimize resource usage	✓		
26.2 Agent performance profiling	✓		
26.3 Optimize code	✓		

Terms

AV	Autonomous Vehicle
SOTA	State Of The Art
AD	Autonomous Driver(s)
SE	Simulated Environment
RL	Reinforcement Learning
LLM	Large Language Model