Working Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Start of the week	18-Nov	25-Nov	2-Dec	9-Dec	16-Dec	23-Dec	30-Dec	6-Jan	13-Jan	20-Jan	27-Jan	3-Feb	10-Feb	17-Feb	24-Feb	3-Mar	10-Mar	17-Mar	24-Mar	31-Mar	7-Apr	14-Apr	21-Apr	28-Apr	5-May
			2/12 Kick - off	**9/12 Discussion on Brain**	**16/12 Discussion on Embedded 18/12 receive car kit 19/12 Discussion on Computer**																				
Sensing and input working package	Documentation on the given guides and projects. Chose main languages and technologies Create/adapt project plan Members tasks asignation	Ubuntu 20.04 & Ros Noetic Study and document vehicle architecture Create/adapt project plan Study given start-up code		Team photo submission Research ROS, sensors, and initial tools Study and document vehicle architecture Plan GitHub repository structure Create/adapt project plan Study given start-up code	Camera handling, preprocessing, noise cancelling, ROIs definition							Define other necessary sensors, define use-case, integration (IMU, distance), preprocessing, noise cancelling.													BFMC
								Define use-case and test given servers information (localisation on map, cars interaction, gps interaction)																	
																		Induce noise on all sensors and systems				Other functionalities and optimizations			
Perception and scene understanding working package					Lane detection			Intersection detection			Traffic sign detection			Traffic light detection											
												Position fusion						Traffic lights detection & classification							
														Define objects properties file				Object detection & classification							
														Environmental server interaction											
																						Other functionalities and optimizations			
Behaviour and motion plan working package					Define project architecture and communication between packages					Define path planning and validation				Define robustness and safety measures											
										Define decision making --> priorities of actions and state flow															
																		Induce noise on systems to valdiate robustness (loss of image, burned image, road search, undefined objects and states)				Other functionalities and optimizations			
Vehicle control working packages					Lane following and speed control					Intersection navigation				Simple action taking maneuvers (parking, stop for traffic sign, stop for traffic light, stop for pedestrian)				Complex action taking maneuvers (swith lane for static and mobile car, road search)							
																						Other functionalities and optimizations			
Final result & Demo				Team can control the physical car remotely and the virtual car on the simulator.		Car can keep a lane, can make a curve				Car can navigate in intersection				Car can go on a pre-determined path, stop at stop sign, park at parking sign, slow at crosswalk				While detecting and calculating it's position, the Car can dynamicaly go to specified checkpoint, react to traffic lights, interact with other cars and send environemt data)							
				Team defines and creates it's own physical testing environment				Team defines a way of parallel developing and testing																	
				Team installs the virtual testing environment																					
																						Other functionalities and optimizations			
Deadlines				**16-Dec**					**20-Jan**				**17-Feb**				**17-Mar**					**21-Apr**			**21-May**
Checkpoint				1st report					2nd report		**Lunar new year**		3rd report				Mid-term quality gate					4th report			**5th report**
Requirements				The team should at least control the car with the given start-up code.					The team should at least link the input data to a rough output (for example, camera to motors).				The team should have at least shown some in-depth algorithmic approaches (for example, show a pretty solid lane-keeping)				The team should have at least some autonomous features ready (for example, show the car keeping its lane and reacting to some signs or obstacles)					The team should show autonomous features almost complete (for example, show specific reactions to particular cases: fog, roundabout).			**Team should show autonomous features complete (car can react to any obstacle on the map).**