Team Name: Domo Arigato
Robot Name: Chipotle 1

Team Members:
Jason DiSalvo
Brian Eckerly
Keun Young Jang
Neal Mehan
Arun Rajmohan

Accomplished So Far

- Familiarized ourselves with startup procedures for the robot
- Familiarized ourselves with various Linux commands – how to navigate through files
- Learned how to maneuver the robot through manual controls
- Learned how to run simulations using Gazebo

Chipotle

Accomplished So Far (cont.)

- Experimented with the LIDAR sensor
- Experimented with the GPS readout of robot position
- Learned how to create the makefiles along with compiling C++ programs
- Learned how to load the programs onto the robot so the robot can run them

Accomplished So Far (cont.)

- Loaded example turn and waypoint navigation program and ran them from the robot itself
- Analyzed both codes and are working on building off of them
- Recorded all our pertinent information learned about the robot and project in an open office document for easy access and use

Schedule

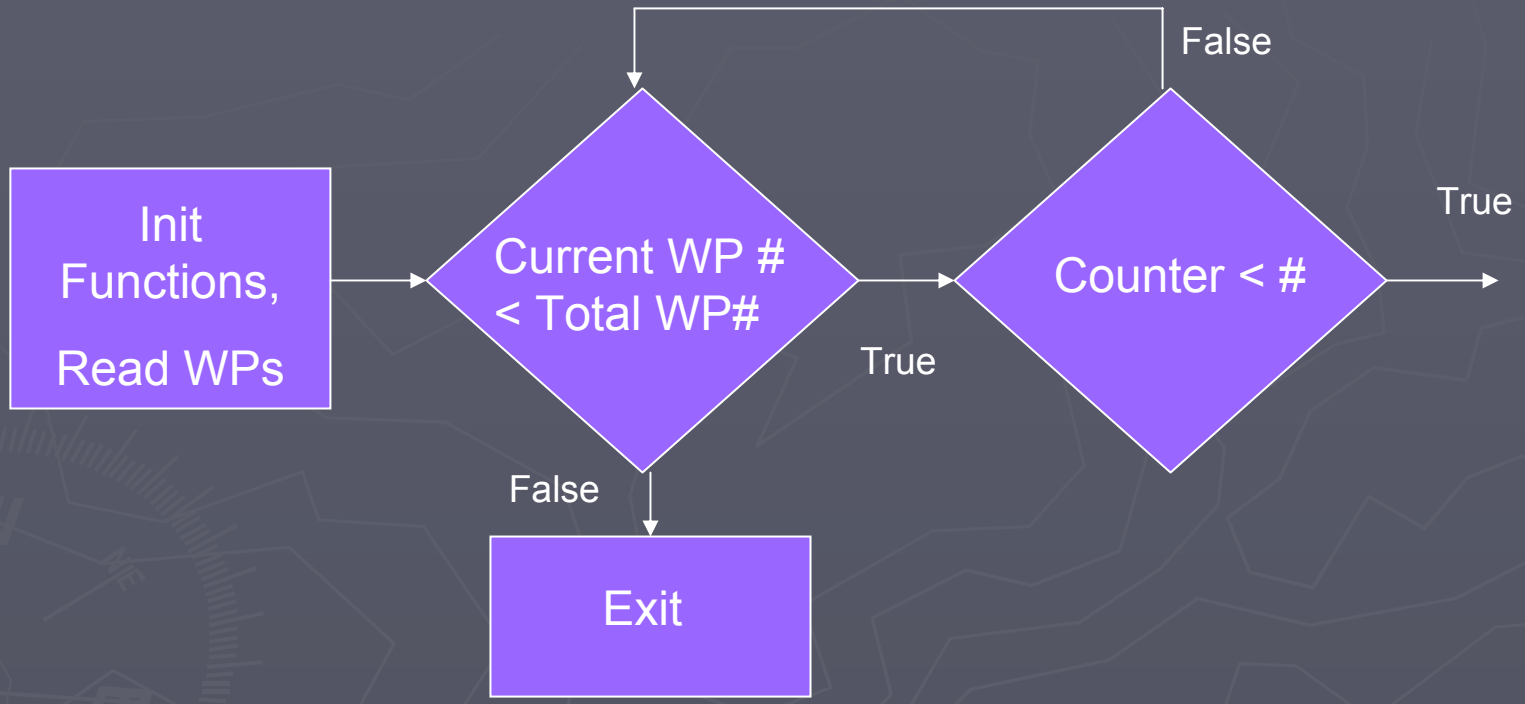
ID	Task Name	Start	Finish	Duration
1	Week 4 Further testinig the player (One point to one point & fallowing waypoints) Specify software coding	1/22/2007	1/26/2007	5d
2	Week 5 & 6 Determining the strategy for the Testing obstacle avoidance (Architecture of software and specifications of coding)	1/29/2007	2/9/2007	10d
3	Week 7 Testing at intersection with simple examples	2/12/2007	2/16/2007	5d
4	Week 8 Testing at intersection with more complex examples	2/19/2007	2/23/2007	5d
5	Week 9 Practice with mission and fixing errors	2/26/2007	3/2/2007	5d
6	Week10 Final Demonstration	3/5/2007	3/9/2007	5d

Next Steps

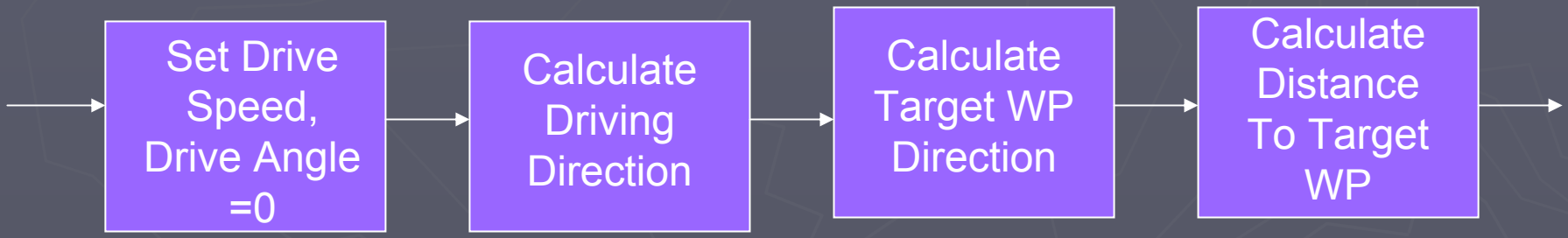
- Develop new waypoint following algorithms (one point to one point and sequence of points)
- Determining obstacle avoidance specifications: where obstacle is located and how define next movements in terms of angle, speed, etc.

Chipotle

Control Algorithm – High Level



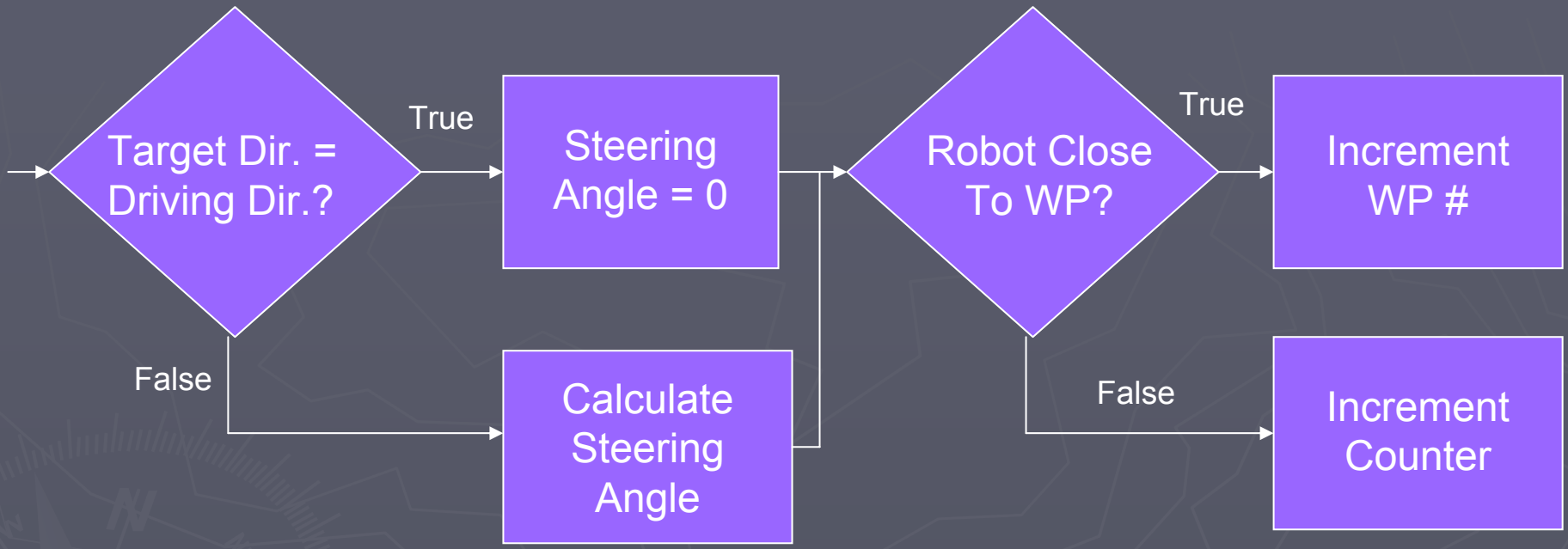
Control Algorithm – Steering Control



- Simple Algorithms used to calculate position data
- Distance to target used as a steering direction weight

Chipotle

Control Algorithm – Steering Control



- Driving Direction, Target Direction, and Target Distance used to calculate steering angle
- Steering Angle does not equal target direction

Chipotle

Obstacle Detection

- First step in obstacle avoidance is obstacle detection…
- LIDAR
  - **Li**ght ***D****etection* **A**nd **R**anging
  - A light sweeps back and forth, measuring which angles produce reflections back to the internal sensor.
  - At the angles where reflections are received, distance is computed by the time it takes the light to return.
  - If we know where the obstacle is in relation to the robot, we have the first step in avoiding it.
- GPS
  - Is used to determine absolute locations of obstructions.
  - If we know absolute location of robot and relative location of obstruction, we know absolute location of the obstruction.
  - May be utilized for some algorithms that plot intermediate waypoints.

Chipotle

Stop!

- First step in obstacle avoidance will be to stop when an obstacle is too close for comfort.
- Wait a fixed period of time.
- Is the object still there? (If it is a car, it might have moved out of the way already.)
- If obstacle is gone after wait time, continue following waypoints.
- If obstacle is still present, a more complicated approach needs to be taken.

Approach #1: Zig-Zag

- Based on LIDAR results, have robot turn in place until there is enough clearance for a straight line path.
  - (direction with least amount of turning required)
- Drive forward a fixed distance until robot is past the obstacle.
- Continue toward the waypoint that was desired before the obstacle was detected.
- If another obstacle is encountered, repeat.

Approach #2: Curve Fitting

- One algorithm that could be used would be to plot intermediate waypoints on a curve.
- This would allow a more realistic motion.

Position and Velocity Algorithm

Another algorithm would be a real-time object following algorithm that continuously monitors the LIDAR and velocity vector (velocity comes from GPS) and adjusts position and velocity accordingly to avoid the obstacle.

Chipotle

Reverse Turning Algorithm

- In order to be realistic with a car, which cannot turn while stationery, the robot can reverse turn until the object is no longer in its forward path and then move past the obstruction and continue on to the way point.

Testing Algorithms

- Our way of testing the algorithms is to first start with a stationary object in a linear path of the robot.
- Next we will test the algorithms with a moving object.
- Lastly we will pick a non-linear path and repeat the two above scenarios.

Chipotle