

## Coordination

- drone\_poses: std::vector<Eigen::Matrix4f>
- motion\_commands: std::vector<Eigen::Matrix3f>

- + Coordination()
- + ~Coordination()
- + assign\_pose(int, Eigen::Matrix4f): void
- + generate\_motion\_command(int): Eigen::Vector3f
- + update\_pose(int, Eigen::Matrix4f): void
- + ensure\_coverage(): void
- + log\_details(): void

## Reconstruction

- individual\_point\_clouds: std::vector<open3d::geometry::PointCloud>
- unified\_point\_cloud: open3d::geometry::PointCloud
- + merge\_point\_clouds: open3d::geometry::PointCloud
- + process\_unified\_point\_cloud: open3d::geometry::PointCloud

- + Reconstruction()
- + ~Reconstruction()
- + add\_point\_cloud(open3d::geometry::PointCloud): void
- + apply\_sensor\_fusion(): void
- + log\_details(): void

## DataCapture

- assigned\_region: Eigen::Vector3f
- waypoints: Eigen::Vector3f
- depth\_data: std::vector<float>
- + capture\_depth\_data(): std::vector<float>

- + DataCapture()
- + ~DataCapture()
- + assign\_region(Eigen::Vector3f& region): void
- + plan\_navigation(std::vector<Eigen::Vector3f>& new\_waypoints): void
- + process\_depth\_data(std::vector<float>& depth\_data): void
- + log\_results():void