# Software Development Process (SDP)

## Principles

- We are responsive and answer any questions and communications within 24 hours.
- We are proactive and complete assignments in advance when possible.
- We are open and transparent when communicating. All ideas and views are treated equally.
- We use Github Projects to manage our backlog, and work on it when possible.
- Backlog should be updated regularly based on discussion during weekly meetings.
- All work items are as small and as granular as possible.
- All changes are developed in a separate git branch.
- After the change is complete, make a pull request and link the issue. Make sure the pull request aligns with the specified definition of done.
- Be open and communicative about any progress made, if there are any delays, make sure to notify team members.
- Don’t be afraid to ask for assistance
- Members need to check that all major functional merges work as expected, even if they pass the pull request.
- All changes to the code should be thoroughly documented once complete.

## Process

- Team Meetings and Planning(1/week)
- Demo/Review with TA(1/week)
- Meeting with Project Partner(~1/ 4 weeks)
- Merge Meeting(1/ 2 weeks)

# Roles

The following roles will be rotated on a monthly basis and tracked using the roles.md file in the repository.

**Administrative Roles:**
Team Lead - Keeps track of due dates, makes sure that team is on track
Liaison - In charge of contacting Project Partner, professors, and T.A.s with any questions
Planner - Sets date and time for weekly meetings, makes an agenda for them
Secretary - Takes notes on meetings

**Development Roles:**
Physics/Navigation Development - Making the movement and collision system
ADAS Designer - design and develop the ADAS implementations
Modeler - In charge of finding and/or making the 3D models for use.
Graphics Designer - Chooses colors, lighting, and potentially shaders to use for the game

# Tooling

| | |
|---|---|
| **Version Control** | GitHub |
| **Project Management** | GitHub Issues and Projects |
| **Documentation** | https://github.com/withastro/starlight, README |
| **Test Framework** | Unreal Engine Test Automation. |
| **Linting and Formatting** | Prettier |
| **CI/CD** | GitHub Actions |
| **IDE** | VS Code |
| **Graphic Design** | Lucid Charts |
| **Others** | N/A |

# Definition of Done (DoD)

- Functionality is complete
- Functionality passes test at an acceptable level
- Changes are merged to the main branch and tested against existing features
- No regressions, entire project functions to spec
- Documentation is updated
- Changes are noted and demoed for next stakeholder meeting
- Backlog updated

# Release Cycle

- Update every successful pull request
- Automatically deploy to staging for every merge to main branch
- Deploy to production every release
- Release every three months
- Use semantic versioning MAJOR.minor.patch
  - Increment the minor version for new features
  - Increment the patch version for bug fixes
  - Increment the major version for breaking API changes

# Environments

| Environment | Infrastructure | Deployment | What is it for? | Monitoring |
|---|---|---|---|---|
| Production | Github Environment | Release | Sleeping well at night | Sentry |
| Staging (Test) | Render through CI/CD | PR | New unreleased features and integration tests | Sentry |
| Dev | Local (macOS and Windows) | Commit | Development and unit tests | N/A |