

# Software Development Process (SDP)

[Principles](#)

[Process](#)

[Roles](#)

[Tooling](#)

[Definition of Done \(DoD\)](#)

[Release Cycle](#)

[Environments](#)

## Principles

- Maintain timely responsiveness on communication channels, aiming to check platforms a minimum of twice daily, ensuring messages are addressed within 24 hours.
- Work consistently to avoid sprints by using the kanban board on the GitHub project for backlog management and ongoing issues.
- All development / debugging will happen in those designated branches of the repository, strictly avoid modifying the main branch directly.
- Each team member needs to have at least one Pull Request.
- New code and major changes that get added to the Github project must be reviewed and approved by the rest of the team before it gets merged with the rest of the code.
- Focus on making a working product before looking into experimental / complex features
- The pull request must meet the definition of done before being moved to “Done” in the kanban board
- All changes to the code must be added through a separate branch. It must not be added through the main branch.
- Work items should not take an excessive amount of time to complete (more than 2-3 days)

## Process

Asynchronous: Most of the time work will be done on our own. Communication will primarily occur online through Discord amongst ourselves and through Teams/email for conversations with the TA and Project partner.

**Planning and Management:** Ideally at the beginning of the week, we will meet either in person or virtually to discuss priorities, define objectives, and split up tasks so that everyone has something they can contribute and report in the weekly standup meetings with the TA.

**Design:** Work items from the “To Do” column will be moved through the workflow as team members make progress.

**Demo/Review with project partner:** Every two weeks (minimum) the team will meet with the project partner to showcase progress, gather feedback and make necessary changes. Project partner feedback will help us organize our priorities.

## Roles

All roles in the team will be very loose. Everyone will be responsible for some level of every task, be it project/product management, design, or code.

## Tooling

<b>Version Control</b>	GitHub
<b>Project Management</b>	GitHub Issues and Projects
<b>Documentation</b>	Github ReadMe <a href="https://github.com/jamesonyee/AR-physics-game">https://github.com/jamesonyee/AR-physics-game</a>
<b>Test Framework</b>	Quest 3
<b>IDE</b>	Visual Studio, Visual Studio Code
<b>Graphic Design</b>	Unreal Engine, Blender

## Definition of Done (DoD)

- Release notes updated
- Acceptance Criteria are valid
- Changes are merged with current codebase
- Integration tests successful

- Changes don't break or cause problems in the simulation
- No significant errors

Minimum viable product?

## Release Cycle

Our product will not be publicly available during development, but we will use a versioning format:

- MAJOR.minor.patch format
  - Increment patch for bug fixes
  - Increment minor for new features
  - Increment major for substantial system changes
- Major versions released every ~3 months (once a term)
- Minor after a new feature is completed
- Patch for bug fixes from previous releases

## Environments

Environment	Infrastructure	Deployment	What is it for?	Monitoring
Production	Quest 3	Release	Final product	N/A
Staging (Test)	Unreal Engine	PR	New unreleased features and integration tests	N/A
Dev	Unreal Engine	Commit	Development and unit tests	N/A