

# 1DL251 SEPM TEAM-B: D1 - PROJECT DELIVERABLE

Date: 03/02/19

## Development team and the skills/strengths they can bring to the team:

- Enzel, Viktor (VE) - good with Python and Git
- Gaide, Maxime (MG) - good with prototyping
- From, Gustav (GF) - good with validation/testing
- Ingvast, Pelle (PI) - good with Python
- Kerle, Laurin (LK) - good with organizing and planning
- Ross, Adam (AR) - good in Python, git & team software development

## Primary development tools:

Communication: Slack & email	- PI is responsible for Slack
Task management: Trello	- GF is responsible for Trello
Team meetings: Doodle	- LK is responsible for Doodle
Designing: draw.io	- MG is responsible for draw.io
Development: GitHub (platform), PyCharm (IDE)	- AR is responsible for GitHub
Accounting & marketing: Uppsaline website	- VE is responsible for Uppsaline accounting

## Further information on team member responsibilities:

Each team member shares responsibilities for all software development roles/tasks such as planning, designing, development, testing, reviewing, and deployment of the game engine for each sprint. What each developer does in respect to each development role per sprint/story/task can be decided during the planning for said sprint/story/task.

All developers should be involved in the sprint planning, and the initial designing for each sprint, and should all be involved in some degree in all aspects of the software engineering development project when required to and should have completed every type of role at least once during the period of the course.

There should be no specific, sole management-related role for any single member of the development team throughout the project. There is already a PO, scrum coach, lecturer and course coordinator for this. Management of the project should be a collaborative effort by all team members.

## Team-B rules:

All rules come into effect following the date and time of this agreement submission.

No member can be held accountable for not following any rules agreed upon prior to the commencement of this agreement.

All team members will be included in any email correspondence with anyone (scrum coach, PO, potential market opportunity, etc.) relative to the development project.

## Sprint rules:

- Each sprint team meeting must be held within the first three days of the sprint and another within the first two days following a client meeting if the client meeting follows the first sprint meeting.
- Sprint planning and design (documentation, such as user requirements and stories, their estimations, system requirements, UML designs, etc.) must be completed by team members collaboratively during the first sprint meeting (or second if there is a client meeting following the first sprint meeting) for each sprint prior to any story development commencement. These tasks - aside from story estimations - can be split between the team or completed by the entire team. In the case of UML diagrams - as they require time, skill and can only generally be designed by one or two people at a time - and any other such documentation tasks dependent on the completion of the requirements tasks be at least addressed and team member(s) made responsible for the completion of each task within a reasonable time-frame for the prompt commencement of following dependent tasks, such as user story development tasks. However, any potential UML design is preferable if completed during the meeting.
- Each such meeting should be planned to be at least two hours long.

- Sprint stories (stories decided to be completed during the sprint) will be collaborative with at least two members responsible for each story implementation (not including the story review).
- Sprint story development will have its own git branch which will be named 'Story#', where # is the number of the story
- Sprint tasks should be in an ascending order of importance in development (which is in a descending order of development process) with dependencies being at top of the list
- Members will make themselves responsible for a single sprint story and a single task for that story during the sprint meeting(s); they will add themselves as members for said task and move the task to the 'Doing' list on Trello.
- A member can only be responsible for one story at a time. Generally, also only one task in that story, unless there is a delay in another task worked on priorly, or two tasks are very closely interrelated, etc.
- Designing for (and prior to implementations of) an individual story and respective tasks will be collaboratively completed by the members responsible for that story either during the sprint meeting(s) or in their own follow-up meeting(s) unless collaborative designing is required across multiple stories, then the members responsible for all these stories will participate in the designing for these stories (ie. the three AI difficulty stories).
- A member is free to pick-up on any available development task in a 'To Do' for the story they are responsible for (or in 'To Do (general)') they are responsible for. However, be aware of dependencies for each task requiring being completed first.
- If a member has completed a story they are responsible for, they may search for available ('To Do') tasks in another story. The other members responsible for that story, if any, must be notified prior to commencement of that task to discuss the current story plan. Beware of story dependencies.
- When a development task for story (or a development task in 'To Do (general)') is completed it must be moved to the 'Pending Review' list.
- When a task is moved to the 'Pending Review' list a comment must be made on the Slack channel #Reviews with a link to (or a description of) the relative development git commits and/or class being developed, and if anyone is already responsible for the review then an @member should be added to the slack comment.
- The reviewing task of any story (or general task) cannot be completed by any member that worked on that task, or any other tasks in the same story of that task.

- If the review of a task results in the task not meeting the specified requirements, the fixing of the task must be completed by the member responsible for the task and not the reviewer.
- If the review of a story or general task is successful then, if there is a branch for the story, the reviewer should merge the branch to the master repository, and all relevant task(s) to the reviewing should be moved from 'Pending Review' to 'Done', and the review task should also be moved to 'Done' on Trello.
- When a task is moved to 'Done' in Trello there should be a link attached to the task for the relevant commits relating to the completion of the task.
- If a branch merge is completed by a reviewer of a story, then a link to the commit for that merge should be attached to the review task on Trello.

#### Team work:

- During Meetings members who are behind on their schedule can ask for help. If they don't do this or are constantly behind, the rest of the team can give a warning. After three warnings a vote can be cast to expel this person. Every other member must vote yes for this vote to be successful.
- Every team member is required to attend at least one team meeting per week aside from the scrum and TA meetings. However, if there is a meeting relevant to the sprint planning, every member implored to attend as this is where the sprint story and task (who is responsible for what) is planned.
- Punctuality is a priority at all pre-arranged meetings, whether if it's a full team meeting, or a portion of the team.
- The team aims to attend all activities with a minimum of 80% of the members.
- Team meetings will either be decided during the prior team meeting or by the team member responsible for organizing team meetings. Each team meeting must be at an appropriate time that suits all team members. Each team meeting time and place must be made available to all team members in a comment on the #meetings channel of the team slack workspace. The comment must be made within a reasonable amount of time prior to the meeting by the team member responsible for team meetings planning. Each member is expected to respond to the doodle within a reasonable time-frame.

#### Git, python and general development conduct:

- Always 'git pull' before you 'git push'.
- Keep git commit headers brief and use imperative grammar (add, not added)
- Include in each commit header either '#implement', '#refactor', '#fix' to define the development (programming) pushed to the git GameEngine repository.
- For peer programming, which is implored, include in each commit a 'Co-authored-by: "user name" <GitHub account email>'. Refer to this webpage: <https://github.blog/2018-01-29-commit-together-with-co-authors/> or use this for a simple work around: <https://desktop.github.com/features/co-authors/>
- Peer programming/task work: each member should work with every other team member at least once on a specific, separate task. That is, complete at least five tasks each, each task being with at least one other team member that you haven't worked with on any other prior task
- Pylint correctness (PEP8 style - PyCharm will always hint with squiggly underlines if the code is not Pylint correct; 1 empty line between methods, 80 chars max per line, etc.) There is also a website for checking pylint correctness: <http://pep8online.com>
- Write descriptive docstrings for each class and method implemented
- Write inline comments for each significant line of code
- Implement classes in separate files
- Follow pythonic class, method, file and variable naming conventions, such as ClassAwesome, a\_method\_example(), player\_ai.py, a\_var, etc.
- Always update in Trello when you are working on a task and when completed a task

#### Slack code of conduct:

- Always write messages in the appropriate channels
- #Assignments are either for the tasks or for the deadlines in the course opinions?
- #Random is for fun/non-work-related stuff
- #Meetings is for meetings. Never write something that isn't a scheduled meeting here. If you must discuss something in this channel do it in replies. This is to improve readability and have an easy way of checking up on dates.
- #General is for everything general questions or requests for tasks, help or discussion.
- #Python is for more technical issues such as not getting Pylint to work. IE not code related
- #Git is also for more technical issues relative to git.
- Just to reiterate only #Meetings have strict restrictions on what/how to write.

#### User story estimations:

- The entire team should be involved in a team agreement for each user story estimation
- User story estimations should be decided using a point system. 1 point represents 1 hour and the lowest amount of points possible to place on a task is 0.5 points.

#### Issues between team members:

- If there is ever any issue (such as unresolvable disagreement on task work, etc) occurring between any team members then, if this is not a matter to be addressed by the University, there should be a meeting held with all team members present to best attempt to resolve the issue.