Odd Space

Production Plan

Chaotic Evil
Òscar Canales, Raul Cano, Denis Deconinck, Carles Garriga, Alex Gesti, Pol Pallarés

Índex

General Calendar of Work

A general calendar that shows every important event in the development process. It includes deadlines, meetings and any major occurrence along the making of the project.

- Sprint: weekly meetings that show what has been done in that week, report bugs and problems inside the team, and define what has to be done for the following week.
- QA sessions: quality assurance sessions in which team members and external testers will play the game and make a report on which aspects of it don't work and what changes should be done.

February

Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun
01-07	-	-	-	-	-	-	-
08-14	-	-	-	-	-	-	-
15-21	-	Start Preproduction / Sprint	-	-	-	-	-
22-28	-	-	Sprint	-	-	-	Sprint

March

Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun
01-07	-	Sprint	-	-	-	-	-
08-14	-	-	-	-	Sprint	END: Prod. Plan / TDD	Sprint / Concept Discovery Deadline
15-21	-	Pitch Presentation / Finish Preproduction	Start Production	-	-	Sprint	-
22-28	-	-	-	-	-	Sprint	-
29-31	-	-	-				

April

Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun
01-04				-	-	Sprint	-
05-11	-	-	-	-	QA session	QA session / Sprint	-
12-18	-	-	-	-	-	Sprint	Vertical Slice Deadline
19-25	-	-	-	-	-	Sprint	-
26-30	-	-	-	-	-		

May

Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun
01-02						Sprint	-
03-09	-	-	-	-	QA session	QA session / Sprint	-
10-16	-	-	-	-	-	Sprint	Alpha Deadline
17-23	-	-	-	-	-	Sprint	-
26-30	-	-	-	-	QA session	QA session / Sprint	-
31	-						

June

Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun
01-06			Sprint	Gold Deadline / Finish Production	Start Postproduction	-	-
07-13	-	-	-	-	-	-	-

14-20	-	-	-	-	-	-	-
21-27	-	-	-	-	-	-	-

Record of the Meetings

A record of every sprint and meeting that takes place within the team.

Meeting 01 - Sprint

- Date: 16/02/2021
- Outcome:
    - Brainstorm of ideas for game's story, art, plot and mechanics
    - Game view orientation decided: Top-Down
- TODO:
    - Develop some of the ideas in depth
- Next meeting:
    - Type: unknown
    - Date: unknown
    - Expected outcome: unknown
    -

Meeting 02 - Lead and Producer meeting

- Date: 24/02/2021
- Outcome:
    - Assigning tasks to team members
    - Established predicted dates for meetings
- TODO: nothing
- Next meeting:
    - Type: sprint
    - Date: 25/02/2021
    - Expected outcome:
        - Final brainstorming
        - Present the different documentation to develop
        - Organize tasks again if necessary
        - Establish some deadlines
        - If possible, start General Calendar of Work

Meeting 03 - Sprint

- Date: 25/02/2021
- Outcome:

        - Final brainstorm
        - Documentation to develop presented
        - Tasks assigned
        - New mechanic idea: random quests
        - Final plot twist for the game: you are the bad guy
- TODO:
    - Develop some ideas in depth
- Next meeting:
    - Type: sprint
    - Date: 28/02/2021
    - Expected outcome:
        - Final game idea (not design)

Meeting 04 - Sprint

- Date: 28/02/2021
- Outcome:
    - Presentation of the ideas of the game
    - Choosing of the final idea
    - Changes of the final idea
- TODO:
    - Develop the mechanics, art and gameplay in depth
- Next meeting:
    - Type: sprint
    - Date: 02/03/2021
    - Expected outcome:
        - Changes on some details of the game

Meeting 05 - Sprint

- Date: 02/03/2021
- Outcome:
    - Presentation of the final game idea
    - Final idea is too ambitious
    - Shortened final idea
- TODO:
    - Tech Design Document (TDD)
    - Adjust game plot to last max 30 minutes
- Next meeting:
    - Type: sprint
    - Date: 12/03/2021

    - Expected outcome:
        - Create pitch
        - Check documentation status and set/modify deadlines

Meeting 06 - Sprint

- Date: 12/03/2021
- Outcome:
    - Presentation of map layouts
    - Player and NPCs sprites are temporal
    - Audio Bible doubts solved
    - Overview on progress of each document
    - New item: ron bottle (revives team members)
- TODO:
    - Change the layout of some scenes
    - Finish all documentation before next meeting
- Next meeting:
    - Type: sprint
    - Date: 14/03/2021
    - Expected outcome:
        - Documentation revision
        - Pitch preparation

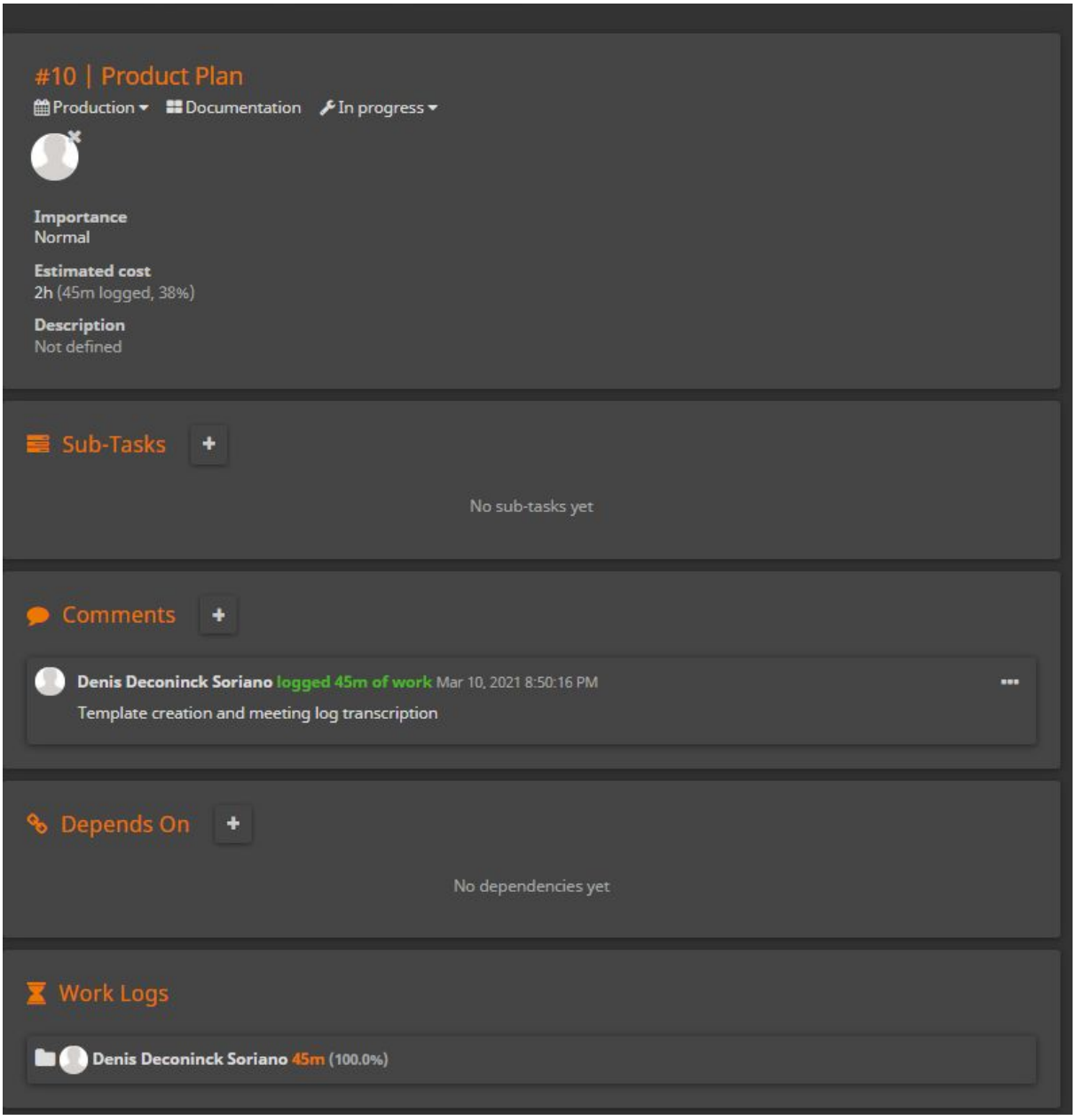
Meeting 07 - Sprint

- Date: 14/03/2021
- Outcome:
    - Finished documentation
    - Started pitch preparation
    - Concept delivery crunch
- TODO:
    - Finish pitch
- Next meeting:
    - Type: sprint
    - Date: 20/03/2021
    - Expected outcome:
        - Start production
        - Define initial tasks for Vertical Slice delivery

Average Estimation Deviation

In order to log how much time we spend developing the game we will use Hack'n Plan. This website lets you indicate the estimated cost of each task (in time) and keep track of how much has already been done.

As an example, the image below shows that the "Product Plan" task is expected to take 2 hours to complete, and has already been worked on for 45 minutes. Moreover, this tool also indicates how much time each member has logged onto the specific task.



After each milestone, the producer, with the help of the rest of the team, will elaborate a document explaining how well time has been managed and how this can be improved.

Risk and Contingency List

Any problem that we may find through the evolution of the game, indicating why it could happen, how it would affect the development process, and how we should proceed in order to solve it.
The template that we are using is the following:

Risk	**Effect**	**Trigger**	**Probability(1-5)**	**Impact(1-5)**	**Contingency Plan**
Definition of the risk	How does it affect the development of the game?	What would make this risk appear?	Probability of the risk appearing	How much would it affect the development?	How can we solve this problem if it appears?

Therefore, we use this for every area (Production, Design, Art and Programming).

Production

Risk	**Effect**	**Trigger**	**Probability(1-5)**	**Impact (1-5)**	**Contingency Plan**
Failure to comply with what was agreed in the sprints or not completing certain tasks	It delays the development process	Thinking we are capable of doing more work than what we can actually achieve	3	3	Move the tasks that we weren't able to complete to the next sprint and be more aware of our capabilities
Not meeting the calendar deadlines	Huge delay on development and probably having to crunch	Being too ambitious with our ideas and not being able to implement them all / last minute bugs and crashes	2	4	Reorganize all the work and, if necessary, cut work on some areas and even crunching

Making a big change without following the team's guidelines	It can lead to miscommunication and team members arguing	Thinking you have a great idea and implementing it without asking the rest of the team	2	4	Make a reunion with the whole team and discuss whether the change is good or should be taken back
Miscommunication inside the team	It could trigger major problems such as not asking for help when needed	Low team connection	4	4	Spending time together doing things out of the university environment, getting to know each other better
Fight between members of the team	Miscommunication between areas (production, design, art, programming), demotivation on the project or decreasing working pace	Differences in opinions and ideas	3	5	Have an intermediary, discuss the problem calmly and reaching an agreement
Someone leaves the team	Less people to develop the game, more stress on the team	The person doesn't show interest in the project, doesn't work as expected, or has discrepancies with other members	2	5	Reorganize all the work, putting more tasks on every member

Design

Risk	Effect	Trigger	Probability (1-5)	Impact(1-5)	Contingency Plan
Quest doesn't make sense in the game's story	Players may not want to complete it and quit the game	Making quests for pure grinding and not justifying them well	2	2	Give the player a special reward or a little NPC dialogue after completing the quest
Ideas are too ambitious	Team is not able to pull of every feature of the game or complete the whole story	Being too motivated and not considering how much time we have to develop the project	5	2-4 (depends on the stage of the game's development)	Deleting some of the ideas, reducing the size of the game and adapting everything to the game's plot
Levels look empty/oversaturated	Development would be delayed as we would have to redesign levels	Not having enough decorations or having too many	1	2	Create more objects or delete some, in some cases this may cause a complete redesign of the level

Puzzles are too hard or easy	Players would get bored or frustrated lowering the game's quality	Designing puzzles without fully testing them	2	3	Test puzzles with people outside the team before approving them
Argumental incoherence	Having to rewrite dialogs and implementing them again	Not understanding 100% the storyline probably due to miscommunication between team members	2	3	Talk with the teammates to find a solution and rewrite the story
Dialogs are too long	Players might get bored and leave the game, so the player base would lower	Being too motivated and wanting to have a big storyline	3	3	Cut some dialog and summarize what we want to say
UI looks confusing or too complex	User would get confused and frustrated so development would have to be delayed in order to reorganize the UI	A lack on knowledge on UI design and not testing it before publishing the game	2	4	Set a meeting with the lead and designer to find a solution and inform the programmers and artist

					about the changes they have to do
Difficulty curve isn't well balanced	Designer would have to do many changes and therefore he would suddenly have a huge amount of work	Not considering well player and enemy stats and how levels affect these	3	4	Change the various formulas that change stats dependin g on levels and reducing the difficulty of some enemies and bosses
AI is too hard/easy to beat	Restructu ring the AI system is hard and would take a long time	Not communicati ng with the programming team and not knowing how to balance the combats correctly	4	5	Stop everythin g on the developm ent, solve the problem on another branch (not the main branch), and when it's solved, continue where we left

Art

Risk	Effect	Trigger	Probability(1-5)	Impact(1-5)	Contingency Plan
Audio volume is too high	Game sounds, by default, would be too loud and that could annoy the player	Not checking loudness before implementing the sounds	4	1	The artist should find a way to lower the volume without distorting the audios
File is too heavy	If this occurs continuously, the game's weight could surpass the established limit of Ram memory usage (256mb)	Having big files with a lot of content on them	2	2	Divide the content of the files in various ones or finding an online compressor that lowers the size of the file without it losing too much quality
Sprite sizes are incoherent/don't match the maps size	Artist would have to change the sprites, and that takes time	Not checking sprite sizes and choosing them just for their looks	2	2	Resize the sprites if possible or find new ones
Typography is hard to read	Dialogs would be illegible and players would not read them	Having a typography that is too small or complex (with serifs)	1	2	Find a new typography, ideally a sans-serif one

Sprites/animations take too long to create	Less time to do other tasks	Thinking we are able to do a big part of the art ourselfs	3	4	Find sprites online to reduce work
Art or music styles are different between scenes	Artist would have to spend time, that we don't have, solving this issue	Using art and audios from different sources that don't match up	3	4	Find new files that blend correctly with what we already have or justify the change with lore
Some element attracts the player's attention unwantedly	That element should be modified by the artist	Having a sprite that stands out in a scene and not noticing it	3	4	Modify the asset with photoshop or replace it with another one
Files have copyright	The game cannot have this types of files, so it would be a major inconvenience if the team doesn't notice it	Not checking if the file is copyrighted beforehand	2	5	If we notice it, replace the file as quickly as possible

Programming

Risk	Effect	Trigger	Probability (1-5)	Impact (1-5)	Contingency Plan
Incorrect art/audio use	It leads to the player being confused	Mistaking which file should be used in a level	2	1	Find in the code where we implement that file and change it
Gamepad problems	Game wouldn't be playable with a gamepad, which is a feature many players like	Not using SDL functions to track gamepad inputs correctly	2	3	Check SDL documentation and asking as much help as needed until it works
Collision detection working incorrectly	It's an essential part of any game, so this would be a huge problem por the development, causing a waste of time and energy	Not having a good collision detection code or not considering tunneling	3	4	Check for online collision detection solutions in c++ and trying to use them as reference
Menu navigation problems	If menu navigation is broken, the game is unplayable	Designing a menu system that's too complex to program or that requires skills that we don't have	3	4	Modifying the menu design or getting more team members to work on it

Save/Load problems	Players will experience problems when saving and loading the game file, frustrating them and resulting in a bad opinion on the game	Not using correctly the pugiXML library to save/load all the variables	2	3	Find where the error is and solve it as soon as possible
Important bug	Game gets broken most probably on an important feature	Trying to implement a new feature for the game and changing something that affects another one	3	4	Work on a side branch in which the rest of the team will keep developing the game while the bug gets solved on the main branch. Ideally, to mark this bug somehow we will use github issues
AI not working properly	Very big waste of time solving the problem	Low knowledge on AI programming or wrong implementation of the AI loop structure defined in the Tech Design Document	4	4	Search for common basic AI implementation in RPG games with code examples and getting all the team to work on this until it behaves as expected

Corrupted files	If a file gets corrupted, it won't appear in the game, making it useless	When saving a file, program crashes and the file gets broken // changing the name of a file that doesn't allow this (maya files for example)	2	5	Make safety copies regularly of all the files to use them in case this happens
Game crashes	The game is not playable	Usually using a null pointer is the cause of this, but there are many other situations that can trigger a crash	4	5	Stop developing the game and center all team members on solving this
Memory leaks	Having this problem makes the ram memory usage increase constantly, even surpassing the limit of 256mb	Not deleting variables from the heap memory	5	5	Finding what variables we are creating and not deleting and delete them when code stops using it