CTIS411 SENIOR PROJECT- 1 SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)

Winged Patrol: VR-based Firefighting Aircraft Simulation Game

Team 17:

Emre Bener
Ömer Faruk Eş
Engin Kaan Görgün
Mustafa Oğulcan Tekiner

Project Supervisor:

Okyay Say

Contents

Table of Figures

1. PROJECT EFFORT ESTIMATION

1.1. Work Breakdown Structure (Decomposition-Based) Estimation

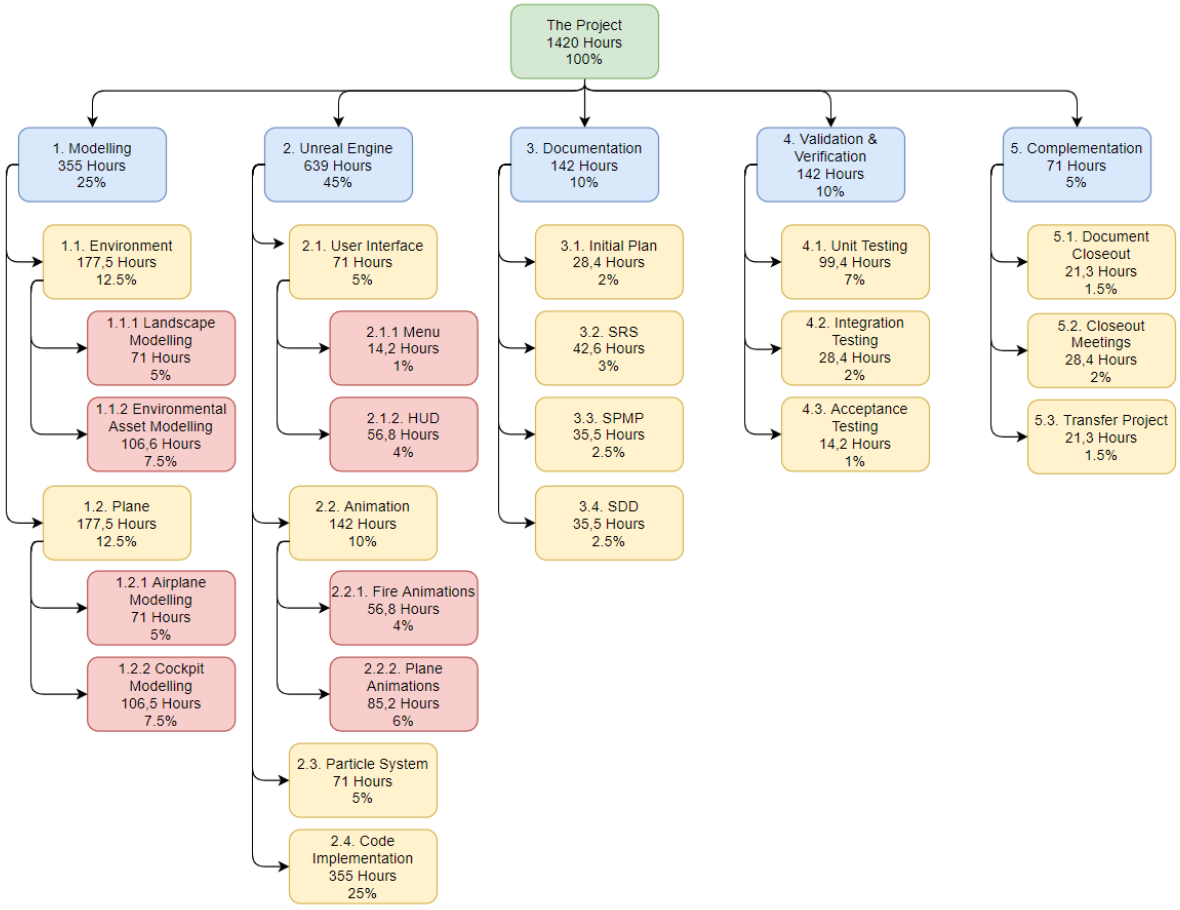

Figure 1: Work Breakdown Structure

1.2. Use Case Based Estimation

Unadjusted Use Case Weight (UUCW)			
Use case complexity	**Weight**	**Number of use cases**	**Product**
Simple	5	6	30
Average	10	4	40
Complex	15	3	45
Total			115

Figure 2: Unadjusted Use Case Weight (UUCW)

Unadjusted Actor Weight (UAW)			
Actor Type	**Weight**	**Number of Actors**	**Product**
Simple	1	2	2
Average	2	3	6
Complex	3	1	3
Total			11

Figure 3: Unadjusted Actor Weight (UAW)

Unadjusted Use Case Points (UUCP)
UUCP = UUCW + UAW
UUCP = 115 + 11 = 126

Adjusting For Technical Complexity			
Factor	**Weight**	**Assessment**	**Impact**
Distributed system	2	3	6
Performance objectives	2	4	8
End-user efficiency	1	2	2
Complex processing	1	4	4
Reusable code	1	0	0
Easy to install	0.5	4	2
Easy to use	0.5	4	2
Concurrent use	2	2	4
Security	1	2	2
Access for third parties	1	3	3
Training needs	1	3	3
Total (TFactor)			36

Figure 4: Adjusting for Technical Complexity

TCF = 0.126 + (0.01 x TFactor)
TCF = 0.126 + (0.01 x 36) = 0,486

Adjusting For Environmental Complexity			
Factor	**Weight**	**Assessment**	**Impact**
Familiar with the development process	1.5	1	1.5
Application experience	0.5	2	1
Object-oriented experience	1	1	1
Lead analyst capability	0.5	3	1.5
Motivation	1	5	5
Stable requirements	2	2	4
Part-time staff	-1	4	-4
Difficult programming language	-1	2	-2
Total (EFactor)			8

Figure 5: Adjusting for Environmental Complexity

EF = 1.4 + (-0.03 x EFactor)
EF = 1.4 + (-0.03 x 8) = 1,16

Putting It All Together UCP

UCP = UUCW x TCF x EF
UCP = 126 x 0.486 x 1,16 = 71,03376
UCP= 71,03376

UCP Final Calculation		
Factor	**Description**	**Weight**
UUCW	Unadjusted Use Case Weight	115
UAW	Unadjusted Actor Weight	11
TCF	Technical Complexity Factor	0.486
EF	Environmental Factor	1,16

Figure 6: UCP Final Calculation

Deriving Duration

Counting the number of environmental factors in E1 through E6 that are above 3 and those in E7 and E8 that are below three. If the total is two or less, assume 20 hours per use case point. If the total is 3 or 4, assume 28 hours per use case. Any total larger than 4 indicates that there are too many environmental factors stacked against the project.

Total is 2 therefore 20 hours per use case.

- The project has 71~ Use Case Points.
- The team will average 20 hours per use case point.
- Iterations will be 2 months long. (Total 3 iterations)
- A total of 4 developers (programmers, testers, DBAs, designers, etc.) will work on this project.

In this case, the complete project will take about **1420** hours to complete (71 x 20 = 1420).

1420 divided by team members (1420 / 4 = 355)
355 hours divided by 6 months (Iteration Cycle X Total Iteration) about 59.
59 divided by 4(four weeks in a month) about 14.75.
Each member will work 15 hours per week in order to complete the project.

1.3. Agile Estimation

Used Method: Power to Two

STORY POINTS				
Priority	**Story Name**	**Coding/Testing Size**	**Complexity**	**Story Points**
High	Start Simulation	Small	Low	1 - Very Small
Low	Options	Medium	Medium	4 - Medium
Medium	Show Credits	Small	Low	1 - Very Small
High	Quit Game	Small	Low	2 - Small
Critical	Control Plane	Large	High	16 - Extra Large
Medium	Fill Water	Large	High	8 - Large
Medium	Release Water	Large	High	8 - Large
Medium	Extinguish Fire	Large	High	8 - Large
Low	Crash Plane	Medium	Medium	8 - Large
Medium	End Simulation	Medium	Medium	4 - Medium
Low	Show Simulation Summary	Small	Low	2- Small

Figure 7: Story Points

Sprint 1	Sprint 2	Sprint 3
1 - Start Simulation	8 - Fill Water	4 - Options
2 - Quit Game	8 - Fill Water	1 - Show Credits
16 - Control Plane	8 - Extinguish Fire	8 - Crash Plane
		4 - End Simulation
		2 - Show Simulation Summary
19	**24**	**19**

Figure 8: Sprint Calculation

2. PROJECT SCHEDULE

Schedule for the First Increment:

	Tas...	Task Name	Duration	Start	End
1		VR Project	167 days	24.09.2020	14.05.2021
2		Documentation	54 days	29.09.2020	11.12.2020
3		Initial Plan	10 days	29.09.2020	12.10.2020
4		SRS	14 days	13.10.2020	30.10.2020
5		SPMP	14 days	3.11.2020	20.11.2020
6		SDO	14 days	24.11.2020	11.12.2020
7		Modeling	62 days	19.11.2020	13.02.2021
8		Environment	60 days	19.11.2020	10.02.2021
9		Landscape Modeling	60 days	19.11.2020	10.02.2021
10		Environmental Asset Modeling	60 days	19.11.2020	10.02.2021
11		Plane	62 days	19.11.2020	13.02.2021
12		Airplane Modeling	25 days	19.11.2020	23.12.2020
13		Cockpit Modeling	36 days	25.12.2020	13.02.2021
14		Unreal Engine	110 days	7.12.2020	7.05.2021
15		User Interface	43 days	16.12.2020	13.02.2021
16		Menu	6 days	16.12.2020	23.12.2020
17		HUD	15 days	25.01.2021	13.02.2021
18		Animation	25 days	1.02.2021	5.03.2021
19		Fire Animations(Extinguish Fire	5 days	8.02.2021	13.02.2021
20		Plane Animations	25 days	1.02.2021	5.03.2021
21		Code Implementation	50 days	8.02.2021	16.04.2021
22		Show Summary&End Simulation	10 days	8.02.2021	19.02.2021
23		Fill Water & Release Water	30 days	16.02.2021	29.03.2021
24		Flight Indicator	30 days	8.03.2021	16.04.2021
25		Control Plane	110 days	7.12.2020	7.05.2021
26		Validation&Verification	125 days	12.10.2020	2.04.2021
27		Unit Testing	100 days	12.10.2020	26.02.2021
28		Integration Testing	15 days	1.03.2021	19.03.2021
29		Acceptance Testing	10 days	22.03.2021	2.04.2021
30		Complementation	12 days	19.04.2021	4.05.2021
31		Document Closeout	3 days	19.04.2021	21.04.2021
32		Closeout Meetings	3 days	26.04.2021	28.04.2021
33		Transfer Project	2 days	3.05.2021	4.05.2021

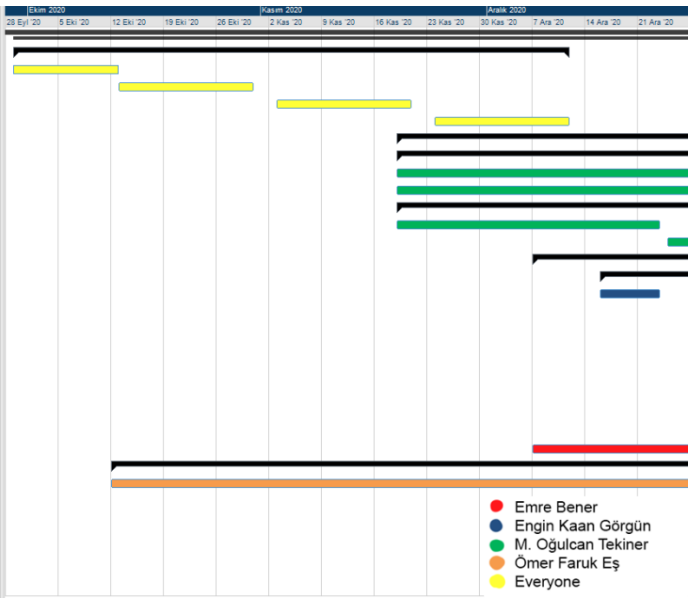

Figure 9: Gant Chart: First Increment

Schedule for the Second Increment:

	Tas...	Task Name	Duration	Start	End
1		VR Project	167 days	24.09.2020	14.05.2021
2		Documentation	54 days	29.09.2020	11.12.2020
3		Initial Plan	10 days	29.09.2020	12.10.2020
4		SRS	14 days	13.10.2020	30.10.2020
5		SPMP	14 days	3.11.2020	20.11.2020
6		SDO	14 days	24.11.2020	11.12.2020
7		Modeling	62 days	19.11.2020	13.02.2021
8		Environment	60 days	19.11.2020	10.02.2021
9		Landscape Modeling	60 days	19.11.2020	10.02.2021
10		Environmental Asset Modeling	60 days	19.11.2020	10.02.2021
11		Plane	62 days	19.11.2020	13.02.2021
12		Airplane Modeling	25 days	19.11.2020	23.12.2020
13		Cockpit Modeling	36 days	25.12.2020	13.02.2021
14		Unreal Engine	110 days	7.12.2020	7.05.2021
15		User Interface	43 days	16.12.2020	13.02.2021
16		Menu	6 days	16.12.2020	23.12.2020
17		HUD	15 days	25.01.2021	13.02.2021
18		Animation	25 days	1.02.2021	5.03.2021
19		Fire Animations(Extinguish Fire	5 days	8.02.2021	13.02.2021
20		Plane Animations	25 days	1.02.2021	5.03.2021
21		Code Implementation	50 days	8.02.2021	16.04.2021
22		Show Summary&End Simulation	10 days	8.02.2021	19.02.2021
23		Fill Water & Release Water	30 days	16.02.2021	29.03.2021
24		Flight Indicator	30 days	8.03.2021	16.04.2021
25		Control Plane	110 days	7.12.2020	7.05.2021
26		Validation&Verification	125 days	12.10.2020	2.04.2021
27		Unit Testing	100 days	12.10.2020	26.02.2021
28		Integration Testing	15 days	1.03.2021	19.03.2021
29		Acceptance Testing	10 days	22.03.2021	2.04.2021
30		Complementation	12 days	19.04.2021	4.05.2021
31		Document Closeout	3 days	19.04.2021	21.04.2021
32		Closeout Meetings	3 days	26.04.2021	28.04.2021
33		Transfer Project	2 days	3.05.2021	4.05.2021

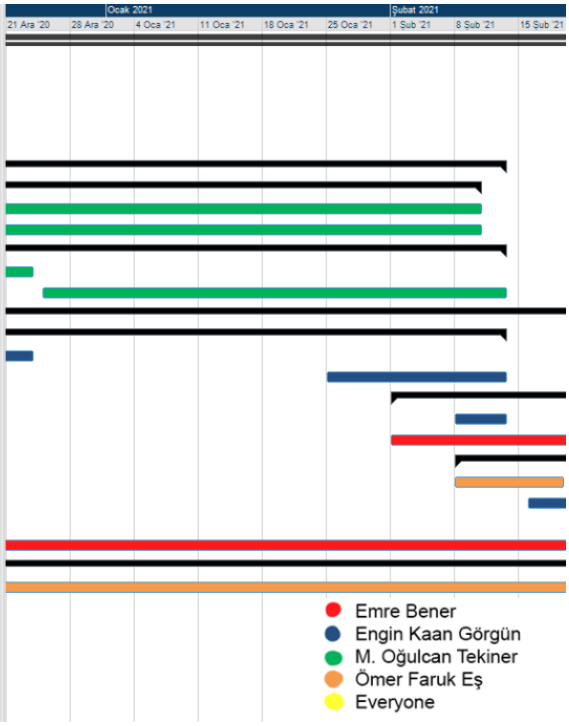

Figure 10: Gant Chart: Second Increment

Schedule for the Third and Last Increment:

	Tas...	Task Name	Duration	Start	End
1		VR Project	167 days	24.09.2020	14.05.2021
2		Documentation	54 days	29.09.2020	11.12.2020
3		Initial Plan	10 days	29.09.2020	12.10.2020
4		SRS	14 days	13.10.2020	30.10.2020
5		SPMP	14 days	3.11.2020	20.11.2020
6		SDO	14 days	24.11.2020	11.12.2020
7		Modeling	62 days	19.11.2020	13.02.2021
8		Environment	60 days	19.11.2020	10.02.2021
9		Landscape Modeling	60 days	19.11.2020	10.02.2021
10		Environmental Asset Modeling	60 days	19.11.2020	10.02.2021
11		Plane	62 days	19.11.2020	13.02.2021
12		Airplane Modeling	25 days	19.11.2020	23.12.2020
13		Cockpit Modeling	36 days	25.12.2020	13.02.2021
14		Unreal Engine	110 days	7.12.2020	7.05.2021
15		User Interface	43 days	16.12.2020	13.02.2021
16		Menu	6 days	16.12.2020	23.12.2020
17		HUD	15 days	25.01.2021	13.02.2021
18		Animation	25 days	1.02.2021	5.03.2021
19		Fire Animations(Extinguish Fire	5 days	8.02.2021	13.02.2021
20		Plane Animations	25 days	1.02.2021	5.03.2021
21		Code Implementation	50 days	8.02.2021	16.04.2021
22		Show Summary&End Simulation	10 days	8.02.2021	19.02.2021
23		Fill Water & Release Water	30 days	16.02.2021	29.03.2021
24		Flight Indicator	30 days	8.03.2021	16.04.2021
25		Control Plane	110 days	7.12.2020	7.05.2021
26		Validation&Verification	125 days	12.10.2020	2.04.2021
27		Unit Testing	100 days	12.10.2020	26.02.2021
28		Integration Testing	15 days	1.03.2021	19.03.2021
29		Acceptance Testing	10 days	22.03.2021	2.04.2021
30		Complementation	12 days	19.04.2021	4.05.2021
31		Document Closeout	3 days	19.04.2021	21.04.2021
32		Closeout Meetings	3 days	26.04.2021	28.04.2021
33		Transfer Project	2 days	3.05.2021	4.05.2021

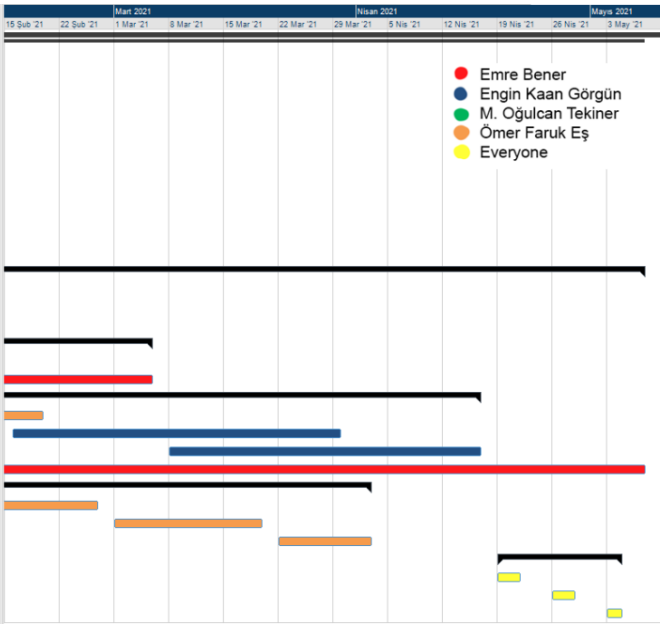

Figure 11: Gant Chart: Third Increment

3. PROJECT MONITORING & MEASURING

3.1. Monitoring Progress

In the development stage of the project, project developers shall use Discord and Zoom platforms to discuss the general status of the project, faced difficulties, upcoming tasks and review of peer works. Meetings will be held on a regular basis such as weekly meetings for discussing general status of the project, deliverables of completed and upcoming tasks. However, there will be some irregular meetings when it is needed. In the meantime, there will be regular meetings with our project supervisor Okyay Say to give feedback on the project and discuss difficulties faced if there is any, for guidance.

General process of the project will be kept track of by using an online Gantt chart which enables us to update the current process dynamically.

3.2. Project, Product, and Process Metrics

Product Metrics		
Metric	**How**	**When**
Performance	Average Frame Per Second(FPS) Average Render Time in milliseconds	Every completed 20% of the project.
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence	After every unit testing.

Figure 12: Product Metrics

Project Metrics		
Metric	**How**	**When**
Length of Code	This is a measure of the size of a program. In general, as the code of a component increases, the component becomes more complex, more open to errors and harder to maintain.	Every completed 20% of the project.
Length of Identifiers	This is a measure of the average length of identifiers (such as names of variables, classes, methods, etc.) in a program. The longer the identifiers, the more likely they are to be meaningful and more understandable.	After every completed task.
Depth of Conditional Nesting	This is a measure of the depth of nesting of conditional statements in the code. Deeply nested if statements are hard to understand and more open to errors.	After every completed task.

Figure 13:Project Metrics

Process Metrics		
Metric	**How**	**When**
Number of Requests for Corrective Maintenance	An increase in the number of bug and failure reports may indicate that more errors are being introduced into the program than are being repaired during the maintenance process. This may indicate a decline in maintainability.	After every unit testing.
Number of change requests	An increase in this number over time may imply a decline in maintainability.	After every meeting with the project supervisor.

Figure 14:Process Metrics

4. PRODUCT VERIFICATION & VALIDATION

4.1. Verification and Validation Techniques

The objective of verification and validation is to assure reliance on a software that matches the requirements of the customers as much as possible. For the product verification purposes, there is a set of orders that are followed by the developers which includes “Requirement Specification”, “High Level Design”, “Detail Design” and “Program Specifications” to make sure we are going to build the product right. For validation, the set of orders that are followed by the developers which includes “Unit Testing”, “Integration Testing”, “System Testing” and “User Acceptance Testing” to make sure we are building the right product.

4.2. Verification & Validation Tools

Verification Tools	
Simulation Model Verification	The models are tested to find and fix errors in the implementation of the models by examining outputs for reasonability. It will be ensured that the topology of the 3D models have no issues such as corrupted topology due to gaps in the model.
Test Cases	Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user.
Unit Testing	In class level, we test functionality of every section of the code to assure the functions work as expected by the requirements.

Figure 15: Verification Tools

Validation Tools	
User Requirements Specification Validation	User requirements as stated in a document called “User Requirements Specification” are validated by checking if they indeed represent the purpose and goals of the stakeholders. This method can be done by communicating with stakeholders directly and/or providing prototypes.
Reviews	Reviews may be used early in the life cycle to provide for software validation.
Simulation Model Validation	The models are tested to check the accuracy of the model’s representation of reality by comparing the model with structural assumptions, data assumptions, and graphical comparison (referencing).
User Input Validation	User input (hardware input) is validated by checking if the input provided by the software operators or users meet the rules and constraints (such as data type, format and latency). It will be ensured that the inputs are working as intended across all supported hardware (keyboard & mouse and VR equipment).

Figure 16:Validation Tools

5. SOFTWARE DEVELOPMENT ENVIRONMENT

Tool	Version	Description
Unreal Engine 4	4.25.4	Real-Time 3D Creation Tool / Game Engine
Microsoft Visual Studio	2019 16.7	Integrated Development Environment
C++	11	Programming Language
Autodesk Maya	2020.2	3D Computer Animation, Modeling, Simulation and Rendering Software
Blender	2.90.1	3D Computer Animation, Modeling, Simulation and Rendering Software
Pixologic ZBrush	2021.1.2	Digital Sculpting Software
Adobe Photoshop	2020 21.2.2.89	Graphics Editor Software
Substance Designer	2019.1.0	Material Creation Software
Substance Painter	2019.1.1	Texturing Software
Gaea	1.2.0.4	Procedural Terrain Creation Software
World Creator	2.4.0	Terrain Creation Software

Figure 17: Software Development Environment