Project plan + study diary

Super Jungle Hunt

version 2.0

Group 3

TUT	Pervasive Computing	TIE-21106 Software Engineering Methodology
Author: Lassi Rintala		Printed: -
Distribution: Lassi Rintala 224895 Pinò Surace 262767 Samu Mäkinen 253582 Vili Saura 240264		
Document status: draft		Modified: 29.4.2018 18:27

VERSION HISTORY

Version	Date	Authors	Explanation (modifications)
1.0	18.01.2018	Lassi R.	Initial version
1.1	29.01.2018	Lassi R.	Added tools & technologies, personnel information
1.2	19.2.2018	Vili S.	Fixing stuff based on feedback
1.3	11.3.2018	Lassi R.	Added sprint 2 study diary
1.4	18.3.2018	Samu M.	Added definition of done, adjusted minor styling errors in Word.
1.5	8.4.2018	Pinò S.	Added sprint 3 study diary
1.6	28.4.2018	Vili S.	Updated extra requirements
2.0	29.4.2018	Vili S.	Added a new header ”Extra features”, renamed the game and added study diary for sprint 4

TABLE OF CONTENTS

1. PROJECT RESOURCES 3

1.1 PERSONNEL 3

1.2 PROCESS DESCRIPTION 4

1.3 TOOLS AND TECHNOLOGIES 5

1.4 EXTRA FEATURES 5

2. STUDY DIARY 6

2.1 SPRINT 1 (RETROSPECTIVE MEETING SUNDAY 11.2.2018) 6

2.1.1 What went well 6

2.1.2 What difficulties you had 6

2.1.3 What were the main learnings 6

2.1.4 What did you decide to change for the next sprint 6

2.2 SPRINT 2 (RETROSPECTIVE MEETING SUNDAY 11.3.2018) 6

2.2.1 What went well 7

2.2.2 What difficulties you had 7

2.2.3 What were the main learnings 7

2.2.4 What did you decide to change for the next sprint 7

2.3 SPRINT 3 (RETROSPECTIVE MEETING 8.4.2018) 7

2.3.1 What went well ... *7*
2.3.2 What difficulties you had ... *7*
2.3.3 What were the main learnings ... *7*
2.3.4 What did you decide to change for the next sprint ... *8*
2.4 SPRINT 4 (RETROSPECTIVE MEETING 29.4.2018) ... 8
2.4.1 What went well ... *8*
2.4.2 What difficulties you had ... *8*
2.4.3 What were the main learnings ... *8*
3. RISK MANAGEMENT PLAN ... 8
3.1 PERSONNEL RISKS ... 9
3.1.1 Risk P1: short term absence of one person ... *9*
3.2 TECHNOLOGY RISKS ... 9
3.2.1 Risk T1: hard disk failure ... *9*
3.3 MANAGEMENT RISKS ... 10
3.3.1 Risk M1: Too low task time estimations causing tight schedule ... *10*
3.3.2 Risk M2: Confusion in task assignment (overlapping implementations etc.) ... *10*
3.4 SOFTWARE RISKS ... 10
3.4.1 Risk S1: Huge refactoring of current implementation ... *10*
3.4.2 Risk S2: Customer changes or adds requirements ... *10*
3.4.3 Risk S3: Minor bugs in the final release ... *10*
3.4.4 Risk S4: Major bugs in the final release ... *11*

1. PROJECT RESOURCES

This chapter holds the project resources.

1.1 Personnel

For each person

- estimate contribution in person hours for each sprint
- travels or other known absences

Product owner:
Lassi Rintala

Development team:
Lassi Rintala (Scrum master for sprints 0 and 1)

- Email: lassi.rintala@student.tut.fi

- Previous experience: 3 years working as a software engineer
- Special skills: C/C++
- Specific fields of interest: Unity

Samu Mäkinen (Scrum master for sprint 3)

- Email: samu.makinen@student.tut.fi
- Previous experience: University coding, slight hobbyist coding for approx. 1 year.
- Special skills: Jack of all trades, master of none.
- Specific fields of interest: Game Design, C++, Unity

Vili Saura (Scrum master for sprint 4)

- Email: vili.saura@student.tut.fi
- Previous experience: High School and University coding.
- Special Skills: C++
- Specific fields of interest: Game Design, Unity

Pinò Surace (Scrum master for sprint 2)

- Email: pino.surace@student.tut.fi
- Previous experience: University study projects
- Special skills: C, C++, Java, Python
- Specific fields of interest: Unity

1.2 Process description

Milestone 1: End of sprint 1, requirements 1-3 done
Milestone 2: End of sprint 2, requirements 4-6 and 10 done
Milestone 3: End of sprint 3, requirements 7-9 done
Milestone 4: End of sprint 4, polishing the game

Goals and success criteria:

- Every member agrees to work around 8 hours per week
- We are aiming for a grade of 4 or higher.

Success measurement:

- Feedback from the customer
- Reaching milestones in time
- Assignment grade

Running the project:

- Meetings twice a week (Sunday evenings physical meeting and a shorter Skype meeting another day)
- Telegram chat group
- Project management with Agilefant
- Version control with Git
- Documentation available
- Scrum master responsibility is changed every sprint
    - Lassi is the first scrum master for sprints 0 and 1
    - Pino scrum master for sprint 2
    - Samu scrum master for sprint 3

    - Vili scrum master for sprint 4
- Responsibilities (who implements what, takes care of what, …) are mostly decided in the weekly meetings
- We discuss the status of the project in our weekly meetings.

Definition of done:
- Agilefant stories are done when all its internal tasks are done and approved upon within the group.
- We agree upon the specifics of each task as a group and play test each change to validate them.

Branching
- We use branching from master to experimental features that we might merge on a later date

1.3 Tools and technologies

Table 1.1: Tools used in the project.

Purpose	Tool	Contact person	version
Documentation	MS Word (word processing) office.microsoft.com	Vili Saura	2010+
	Doxygen (comment notation and documentation generation)	Lassi Rintala	
Communication	Telegram	Pinò Surace	
	Skype (internet calls) http://www.skype.org	Pinò Surace	
Version management	Git	Lassi Rintala	
Code implementation and compilation	Unity	Samu Mäkinen	2017.3.0f3
	Visual Studio	Lassi Rintala	2017

1.4 Extra features

In addition to the user requirements the following extra requirements were added by us:
- Improved scoring: in addition to the speed the player earns score according to their performance (ex. how low on the rope they are willing to jump) They are graded accordingly, and the grade is shown after the fact.
- Difficulty level: In addition to the game becoming harder in every completion, the player can choose their difficulty before they start the game.
- Extra level: After finishing level 4 for the first time, the player is taken into a secret 5^{th} level.

- Randomized New Game +: after finishing the game, the order of the levels is randomized, ex player might start their second playthrough at level 3.
- Minor additions: Sound system with music and level transitions that show the score that the player accumulated during the level.

2. STUDY DIARY

This chapter holds your journal of lessons learned during the course. That is, **more detailed analysis of previous Sprint's contents**.

2.1 Sprint 1 (Retrospective meeting Sunday 11.2.2018)

2.1.1 What went well

- Work load distribution
- Meetings
- Project work in general
- Studying Unity

2.1.2 What difficulties you had

- Modeling the rope
- Player catching the rope
- Difference between 2D and 3D game object components (tutorials in 3D)
- Scaling issues with GUI (canvas, camera)

2.1.3 What were the main learnings

- How to use Git
- Unity basics
- Agilefant basics
- Task estimation
- Agile methods

2.1.4 What did you decide to change for the next sprint

- Scrum master changed to Pino
- Potentially changing the game theme from jungle to something else
- Make all the levels in some template / placeholder form

2.2 Sprint 2 (Retrospective meeting Sunday 11.3.2018)

2.2.1 What went well

- Work load distribution
- Meetings
- Project work in general

2.2.2 What difficulties you had

- Learning Unity
- Technical difficulties with Unity editor
- Overlapping work
- Scaling issues with GUI

2.2.3 What were the main learnings

- More about Unity
- Teamwork
- Agilefant was utilized better this sprint

2.2.4 What did you decide to change for the next sprint

- Scrum master for sprint 3 will be Samu
- Potentially changing the game theme from jungle to something else
- Moved requirement 10 to sprint 2, so sprint 4 is reserved only for polishing the game

2.3 Sprint 3 (Retrospective meeting 8.4.2018)

2.3.1 What went well

- Work load distribution
- Meetings
- Project work in general
- Implementation of extra customer requirement went very well
- Handling of bugs

2.3.2 What difficulties you had

- Unity
- Technical difficulties with Unity editor
- Customer Requirements were added
- Bugs introduced during development

2.3.3 What were the main learnings

- Animation advanced settings
- To be ready for changes in the requirements
- Asynchronous execution in Unity
- Flip of the Sprites in Unity
- Sounds in Unity

2.3.4 What did you decide to change for the next sprint

- Scrum master for sprint 4 will be Vili
- Potentially changing the game theme from jungle to something else
- Change orientation of the game levels

2.4 Sprint 4 (Retrospective meeting 29.4.2018)

2.4.1 What went well

- Work load distribution
- Project work in general
- Handling of bugs

2.4.2 What difficulties you had

- Release process
  - New bugs were found
  - Having the standalone build work on different computers
- Some features and details were not discussed in the group before they were implemented
- Meeting schedules
  - Due to other school work and obligations

2.4.3 What were the main learnings

- Unit tests and other testing should have started earlier in the development process

3. RISK MANAGEMENT PLAN

Table 4.1: Project risks.

Risk ID	Description	Probability	Impact	Seriousness
P1	Short term absence	3	2	6
T1	Hard disk failure	2	2	2
M1	Too low task time estimations causing tight schedule	2	2	2

M2	Confusion in task assignment (overlapping implementations etc.)	1	1	1
S1	Huge refactoring of current implementation	3	3	9
S2	Customer changes or adds requirements	2	2	4
S3	Minor bugs in the final release	3	1	3
S4	Major bugs in the final release	1	3	3

3.1 Personnel risks

Try to estimate risk probability, use a scale of **1 to 3** or Small, Medium, Large.
Other criterion will be the impact or severity. So, how the risk will harm you, if realized. Use similar scaling as in probability.

3.1.1 Risk P1: short term absence of one person

Root cause (source): A key person will be absent for several days.
Importance (seriousness): from the table, basically probability and impact, possibly combined with frequency.
Avoidance: Avoid being near people who have a flu
Response (prevention): Redistribute the work load and share all relevant information, so that the team will be able to carry on.
Recovery (survival): Redistribute the workload; focus on the most important features.

3.2 Technology risks

3.2.1 Risk T1: hard disk failure

Symptom, early warning sign: disk makes noise, arbitrary reading errors occur more often than before.
Source or reason: hard disk is at the end of its lifespan, or hard hit on computer while disk was running.
Probability: 2 medium (on scale 1-3)
Seriousness: 2 medium (on scale 1-3)
How to avoid: buy a new disk when starting a project.
How to prevent: when first symptoms occur, take additional back-ups and change the disk as soon as possible.
How to survive: back-ups, and a replacement disk or whole computer.

3.3 Management risks

3.3.1 Risk M1: Too low task time estimations causing tight schedule

Reason: Tasks take longer to complete than originally estimated
How to avoid: Make estimations always bigger than expected time used
How to prevent: Really concentrate on roughly implementing the feature rather than for example paying too much attention to small details
How to survive: Implement tasks in the order of priority

3.3.2 Risk M2: Confusion in task assignment (overlapping implementations etc.)

Reason: Task status is not updated correctly in Agilefant or otherwise absence of communication between team members about which tasks they are working on
How to avoid: Always keep Agilefant updated
How to prevent: Ask team members if someone is already working on the task you are about to start
How to survive: Choose one of the parallel implementations to be used, discard others

3.4 Software risks

3.4.1 Risk S1: Huge refactoring of current implementation

Reason: Some software component has been first poorly designed and needs reimplementation in a new way to support further development
How to avoid: When designing feature implementations, think about them in their context far ahead
How to survive: Coordinate development so that the refactoring won't cause too much interference in other developers work

3.4.2 Risk S2: Customer changes or adds requirements

Reason: Customer changes existing requirements or adds some more
How to avoid: Can't be avoided
How to survive: Have the software implemented so that adding more things in it is easy enough

3.4.3 Risk S3: Minor bugs in the final release

Reason: Due to implementation not careful enough, bugs are still existing in the final product release
How to avoid: Extensive testing and careful design and implementation throughout the development process
How to survive: Nothing to do after final release

3.4.4 Risk S4: Major bugs in the final release

Reason: Due to implementation not careful enough, bugs are still existing in the final product release
How to avoid: Extensive testing and careful design and implementation throughout the development process
How to survive: Nothing to do after final release