Project plan+study diary

Gikoball

version 1.3

TUT	Pervasive Computing	TIE-21106 Software Engineering Methodology
Author: Thibault Deutsch		Printed: 17.04.2015 13:11
Distribution: Thibault Deutsch, Baptiste Delhommeau, Denis Castéran, José Carlos Ramìrez, Marko Leppanen		
Document status: draft		Modified: 17.04.2015 13:10

VERSION HISTORY

Version	Date	Authors	Explanation (modifications)
1.0	28.01.2014	Marko L.	Initial version
1.1	11.02.2014	Marko L.	Deleted finnish text
1.2	18.01.2015	Tensu	Sections 1.4.x, cosmetic tuning
1.3	26.01.2015	Marko L.	Final touches

TABLE OF CONTENTS

1. PROJECT RESOURCES

1.1 Personnel

Thibault Deutsch, Baptiste Delhommeau, Denis Castéran and José Carlos Ramìrez form the team.

Here is the contact information:

- Thibault Deutsch (249441) – thibault.deutsch@student.tut.fi
- Baptiste Delhommeau (249821) – baptiste.delhommeau@student.tut.fi
- Denis Castéran (249381) – denis.casteran@student.tut.fi
- José Carlos Ramìrez (249641) – jose.ramirezvega@student.tut.fi

Thibault, Baptiste and Denis come from France. They study computer science in EPITA, in Paris. They have good background in programming and have already made two big projects (six months), in groups of four people.

José is from the University of Zaragoza, in Spain, and he study Telecommunication and Networking. He has never worked in a long group project, only one or three months. He has knowledge in Computer Architecture, Distributed Systems and Digital Electronics Systems.

Members of the team are equal. They all share the same responsibility. This allows us to be more flexible in term of organization and more reactive if a risk happens.

1.2 Process description

Our preliminary roadmap for the project is:

- Basic game engine (physics, background, …) and user interface.
- Then, the game logic.
- Then, working on the story and the game experience.
- And at the end, finalize the graphic design and other specifications define by the User Story.

Our goal for this project is to make a Wizzball like game that works and that is fun to play. Moreover, we measure our success criteria by sending to our friend the game and ask them to comment. If 70% of the players think that the game is good, our goal will be reached.

The team tries to organize a meeting each week. During this meeting, we take the time to organize the project and speak about the works

that need to be done. Each member can choose the tasks they want, but we try to split in equal part the fun tasks.

When a task is finished, the developer marks the task as “Ready” in Agilefant. When another member has free time, he can check the code and can test the task. If the task is well implemented, he can change the task to “Done” in Agilefant. We can also do that during the weekly meeting.

All the development process is made in a Git branch called “develop”. When some tasks are done (that means that the tasks were well tested) and if there is no others tasks started in the develop branch, we merge develop into master. With this process, we always have the lasted stable version in our master branch.

When a developer starts a long task, he can create a new branch where he can make all his work without disturbing the others developers. When he has finish, the branch is merge into develop.

If we encounter difficulty or if we want to spend some good time together, we organize a coding night (with beer of course) in an apartment.

Moreover, we use Facebook group message to communicate instantaneously all the time. It’s really fast because members receive notification on their phone, so in case of emergency, we can answer/react as fast as possible.

1.3 Tools and technologies

This project is made using Processing. So, of course, we use the native Processing IDE and Java SDK. Moreover, for all the communication we use Facebook Messenger and Skype (but we prefer IRL meeting).

For the source code, we have created a git repository on Github. The repository is open (because it’s a student project that doesn’t need to be confidential). But only the contributors – the four member of the team – have a write access to the repository. Other people can only read the code source, submit bug via the tracker and make pull-request that need to be validate by the collaborator. The repository is available here: https://github.com/dethi/Gikoball

If a new version of Processing is released during the project development, we have decided to not update our software. This will guard us from all kind of new bugs.

Table 1.1: Tools used in the project.

Purpose	Tool	Contact	versio

		person	n
Documentation	MS Word (word processing) office.microsoft.com	T.D	2011
	Agilefant http://www.agilefant.com	T.D	2015
Communication	Facebook Messenger http://www.facebook.com	All	2015
	Skype (internet calls) http://www.skype.org	All	2015
Version management	Git http://git-scm.com/	T.D	2.3.0
IDE	Processing https://processing.org/	All	2.2.1

1.4 Sprint backlogs

You can find our sprint backlogs on Agilefant.

2. STUDY DIARY

2.1 Sprint 1

During this sprint, we finished the User Story 1 and 2. The description bellow come from the website of Software Engineering Methodology course.

- User story 1:
  - ✓ When the application starts, it will inquire the player's name.
  - ✓ After this, the game greets the player, tells a compelling background story and the game will begin.

- User story 2:
  - ✓ When the game starts, it will show the player character (the ball) in the middle of the 2D play area. The play area consists of floor on which the ball bounces and a background.
  - ✓ The ball can be maneuvered around the screen using the keyboard. Whenever ball bounces up, it may be decelerated with a key press. When the ball is coming down, it may be accelerated with key presses. The ball's velocity while impacting the floor will set the trajectory for the next bounce.

We have already faced an issue. Indeed, Thibault and Baptiste made a trip in Lapland. During two weeks we hadn't the possibility to organize a meeting. The project started a little bit late.

Moreover, nobody could go to the Agilefant exercise. We have learned to use it on our own. So, our backlog is not perfect.

For next sprint, we will first create all the User Story and the task in Agilefant and then decide the planning. Moreover, we will also add to Agilefant all task about documentation, like completing this document. This will help us to organize our time and guard us from a work overload (and maybe also prevent the risk P2). Finally, we will continue to define new possible risks.

2.2 Sprint 2

During this sprint, we finished the User Story 3, 4 and 5. The description bellow come from the website of Software Engineering Methodology course.

- User story 3:
  - ✓ The play area will also have different platforms "in the air" on which the ball can land or bounce from. These are basically just more floor in the air. There is also a "ceiling".
  - ✓ The ball will have velocity and momentum. The ball will accelerate automatically when it falls down due to gravitional acceleration. With a special key, the direction of the gravity can be changed from downward force to upward force and vice versa.
  - ✓ The rebound will be determined based on the angle of impact, elasticity and velocity and the laws of conservation of energy, just as in the real world.
- User story 4:
  - ✓ The play area will scroll when ever the ball approaches the either the left or right border of the screen. The scrolling area reveals more and more of new floor and platforms. There might be holes in the floor and ceiling, allowing the ball to exit the screen from upper or lower border. The game ends if the ball exits from these holes.
  - ✓ The background will also scroll with the level, preferably using some form of parallax scrolling.
- User story 5:
  - ✓ There is a timer counting down seconds from an intial value shown on the screen. When it reaches zero, the game ends.
  - ✓ If the ball reaches the end of the game level (the farthest point in the right of the level) before timer runs out, the player can continue from the next game level.

Exchange student like to travel a lot, so we have faced same issue as in the sprint 1. First, José went in Lapland for one week. And then Thibault went three days in Stockholm.

Moreover, at the beginning of this sprint we defined a new planning process. This process includes a new method to define who do what and a better way to control the quality. This new process works really well.

Next sprint, we will try to have a better time estimation of each task.

2.3 Sprint 3

During this sprint, we finished the User Story 5 (continued), 6 and 7. The description bellow come from the website of Software Engineering Methodology course.

- User story 5 (continued):
  - ✓ There is a timer counting down seconds from an intial value shown on the screen. When it reaches zero, the game ends.
  - ✓ If the ball reaches the end of the game level (the farthest point in the right of the level) before timer runs out, the player can continue from the next game level.
- User story 6:
  - ✓ Some nasties are also infesting the game level, flying, hovering, crawling, standing, bouncing around. If the ball hits a nasty, the game ends.
  - ✓ However, there also are some power-ups around the level, too. With a power-up, the ball can temporarily collide with a nasty, destroying the nasty instead of the ball.
- User story 7:
  - ✓ There also is some special power-ups the ball has to collect before entering the level exit.
  - ✓ Destroying nasties and collecting specialities bestows the player score points which are shown on the screen. Enough points give the player an extra life.

During this sprint, we faced some risk. Risk P2 was the hardest to recover from. All the member of the team had in average two-three new projects in others courses. That's a lot of work and honestly, we weren't prepared for that.

Another minor risk that happened was the risk T3. During one or two weeks, GitHub was the target of a DDoS attack. The website was never offline, but it was really slow.

Moreover, we have defined in this document our process of merging Git branch (master and develop).

Next sprint, we will try to not be overcrowded by all the others projects. Planning will be more than necessary.

3. RISK MANAGEMENT PLAN

Table 4.1: Project risks.

Risk ID	Description	Probability (scale of 5)	Impact (scale of 5)
P1	Short term absence	3	2
P2	Projects for other courses	5	3
P3	Vacation and trip	3	4
P4	Lack of motivation	2	5
T1	Hard disk failure	2	2
T2	Broadband Internet problem	3	1
T3	Github down	1	1

3.1 Personnel risks

3.1.1 Risk P1: short term absence of one person

Root cause (source): a key person will be absent for several days. Someone is absent if we have no response for more than 2 days.
Seriousness: 5
Response (prevention): redistribute the workload.
Recovery (survival): focus on the most important features.

3.1.2 Risk P2: projects for other courses

Root cause (source): members of the team have to work on other projects.
Seriousness: 15
Response (prevention): redistribute the workload and share time between all the projects. Try to avoid rush time.
Recovery (survival): focus on the most important features.

3.1.3 Risk P3: vacation and trip

Root cause (source): members of the team would like to visit Finland.

Seriousness: 12
Response (prevention): redistribute the workload and require members to plan their vacation 2 weeks in advance.
Recovery (survival): focus on the most important features or avoid vacation. Other solution would be to work more during one week.

3.1.4 Risk P4: lack of motivation

Root cause (source): project becomes boring or members don't want to work
Seriousness: 10
Avoidance: have fun during the project, like making coding night or party with the team.
Response (prevention): change the planning and add cool feature that members want to implement.
Recovery (survival): try to motivate the team during a meeting and 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.
Root cause (source): hard disk is at the end of its lifespan, or hard hit on computer while disk was running.
Seriousness: 4
Avoidance: buy a new disk when starting a project.
Response (prevention): when first symptoms occur, take additional back-ups and change the disk as soon as possible.
Recovery (survival): back-ups, and a replacement disk or whole computer.

3.2.2 Risk T2: broadband Internet problem

Symptom, early warning sign: network lag
Root cause (source): operator problem or router failure.
Seriousness: 3
Response (prevention): call hotline or buy a new router.
Recovery (survival): use the Internet connection at the University

3.2.3 Risk T3: Github down

Root cause (source): Github server down.
Seriousness: 1
Recovery (survival): all member of the team have a clone of the repository on their disk, so we only need to push the repository in an other place, like Bitbucket.