Project game - implementation

Kamil Grabowski, Filip Grajek, Bartosz Jasiński, Tomasz Koter, Ivan Rukhavets

Version 1.5

March 6, 2017

Date	Author	Description	Version
Feb 27, 2017	Kamil Grabowski, Filip Grajek, Bartosz Jasiński, Tomasz Koter, Ivan Rukhavets	Initial version	1.0
Mar 3, 2017	Kamil Grabowski, Filip Grajek, Bartosz Jasiński, Tomasz Koter, Ivan Rukhavets	Added project roles	1.1
Mar 4, 2017	Filip Grajek, Tomasz Koter	Added project schedule	1.2
Mar 4, 2017	Filip Grajek, Tomasz Koter, Bartosz Jasiński	Added new specification errors	1.3
Mar 5, 2017	Kamil Grabowski, Filip Grajek, Bartosz Jasiński, Tomasz Koter, Ivan Rukhavets	Added personal work schedule	1.4
Mar 5, 2017	Filip Grajek, Bartosz Jasiński	Added more specification errors	1.5

Contents

1 Specification errors

Id	Location	Remarks	Links
1	Fig. 3.8	How is initial player location determined?	
2	Fig. 3.14	Should distance to piece take into account pieces already being carried by other players?	https://se2.mini.pw.edu.pl/17-results/17-results/issues/61
3	Sec. 1.4	Possible player moves list lacks *pick up piece* action	https://se2.mini.pw.edu.pl/17-results/17-results/issues/62
5	Sec. 2.6	Shouldn't the Game Master also have a –conf parameter?	https://se2.mini.pw.edu.pl/17-results/17-results/issues/63
6	Sec. 3.2.1, 2nd paragraph, list pt. 1	"Game Mastered" typo	https://se2.mini.pw.edu.pl/17-results/17-results/issues/50
7	Sec. 3.2.1, 2nd paragraph, list pt. 1	"send" typo, should be "sent"	https://se2.mini.pw.edu.pl/17-results/17-results/issues/50
8	Sec. 2.5, action delay list	Are actions supposed to be asynchronous or synchronous (ie. can a player request *test* (500 ms) and during that time *move* (100 ms) 5 times?)? We assume synchronous, as it would be pointless to request *discover* and then *move* somewhere else.	https://se2.mini.pw.edu.pl/17-results/17-results/issues/64
9	Sec. 3.2, 1st paragraph	What will happen if Player which carries piece attempts to take another piece by sending PickUp message.	https://se2.mini.pw.edu.pl/17-results/17-results/issues/65
10	Sec. 3.2, 1st paragraph	What info Player will get if he sends Discover request directly alongside Goal Area or when he is on Goal Area.	
11	Sec. 3.2, 1st paragraph	What happens if Player tries to place piece on Task Area.	https://se2.mini.pw.edu.pl/17-results/17-results/issues/66

2 Software development methodologies

The team implements scrum methodology. Every Monday of the semester the team conducts a three-hour long meeting. In the first 15 minutes next sprint is planned, rest of the meeting is intended for coding. Sprints last one week and begin each Monday after the team's meeting. Additionally, the team holds two more meetings a week to discuss

the ongoing process. Other than that the team shall maintain a constant connection via Slack or other messengers.

The team should utilize GitLab's issue board for creating backlogs, planning sprints and organizing workflow.

During meetings the team shall produce following documents:

1. Updated issue board
2. Meeting protocol
3. Backlog (every sprint-planning meeting - Mondays)

According to scrum methodology, team members are assigned following roles:

Product owner Bartosz Jasiński

Scrum master Filip Grajek

Additionally, every team member holds developer's responsibilities. Issue board administration is responsibility of Kamil Grabowski and meeting protocols are responsibility of Tomasz Koter. Any other not predicted responsibilities shall be distributed on the fly.

3 Software technologies

The project is designed in .NET C# using Microsoft Visual Studio. Every member of the team already has two years of experience with this environment and there was no other environment mutual for the whole team, hence the choice was obvious.

4 Schedule

4.1 Project schedule

The whole project can be divided into four main phases. The length of those phases is determined by project deadlines. Every phase has to be ready two days before the given date, the last two days are used to fix bugs found during the "testing" deadlines.

Id	Phase	Estimated time	From	To
1	Communication	16 days	6.03.2017	21.03.2017
2	Game	21 days	22.03.2017	11.04.2017
3	Cooperation	36 days	12.04.2016	17.05.2017
4	Championship	11 days	18.05.2017	28.05.2017

Each phase is divided into smaller tasks, that are assigned man-hours. Those hours

also include time for unit test, which are written after each task.

Phase	**Category**	**Task**	**Man-hours**
Communication	Server	Connecting to the server	15
		Creating game	15
		Joining game	22
		Message flow	22
	Game master	Mock game master	8
	Player	Mock player	8
	Tests	Integration tests	15
	Bugs	Bug fixing	15
Game	Game master	Connecting to server	9
		Creating a game	15
		Accepting players	8
		Board Creation	15
		Data responses	29
		Ending game	8
	Player	Connecting to game	8
		Player messages and actions	29
		Simple strategy	29
	Tests	Integration Tests	8
	Bugs	Bug fixing	8
Cooperation	Integration	Integration of communication server	86
		Integration of game master	80
		Integration of players	79
	Bugs	Bug fixing	8
Championship	Player	Player strategy	50
		Leader strategy	30

4.2 Personal work schedule

Functionality	**Bartosz Jasiński**	**Filip Grajek**	**Ivan Rukhavets**	**Kamil Grabowski**	**Tomasz Koter**
Connecting to server	1	6	6	1	1
Creating game	1	1	6	1	6
Joining game	9	3	2	6	2
Message flow	2	6	2	6	6
Mock game master	1	1	1	3	2
Mock player	4	1	1	1	1
Integration test	3	3	3	3	3
Bug fixing	3	3	3	3	3
Connecting to server	1	3	3	1	1
Creating a game	5	1	4	4	1
Accepting players	3	2	1	1	1
Board creation	1	1	2	5	6
Data responses	2	8	8	3	8
Ending game	3	1	1	2	1
Connecting to game	3	1	3	1	1
Player messages and actions	7	8	3	8	3
Simple strategy	1	1	1	1	4
Integration Tests	3	3	3	3	3
Bug fixing	3	3	3	3	3
Integration of communication server	10	20	20	15	20
Integration of game master	15	10	10	25	20
Integration of players	24	19	19	9	9
Bug fixing	3	3	3	3	3
Player strategy	7	7	12	12	12
Leader strategy	9	9	4	4	4

The numbers in the table above indicate how many hours each team member spends on each task. Most often two to three people work on a task, while the rest is doing another one. Every person has to spend at least one hour on each task to review and understand the code.