Task Assignment

Author: Yilmaz Uzun

Scope	Task	Priority	Description	Assigned
Project Setup	Maven Setup	5	Maven needs to be set up to automatically bulid jar of game, and manage build	Yilmaz
	Project Structure	5	Directories need to exist to divide files later into logical parts	Yilmaz
Network	Server-Client	5	Connection between Client and Server can be established and messages can be exchanged	Yasin & Valentin
	Messages	5	Abstract class Message and many inheriting Messagetypes supporting all needed data transfers	Yasin & Valentin
	Protocol Server	5	Protocol Thread for Server to compute incoming messages	
	Protocol Client		Protocol Thread for Client to compute incoming message	
Game Model	AbstractPlayer	5	Abstract class of player which can interact with game is foundation for every inheriting subclass	Yilmaz
	HumanPlayer	5	Inherits AbstractPlayer, Represents Human player on server-side	Yilmaz
	AbstractAiPlayer	5	Foundation for every Ai subclass, represents Ai player server-side	Yilmaz
	SimpleAi	5	Simple AI, should be able to think and make low-scoring moves	
	SmartAi	5	Smart AI, should able to 'think' and be pretty hard to win against (MinMax alpha-beta-pruning)	
	Board, BoardField, Tile	5	Components of the game, containing essential attributes, getter & setter	Yilmaz
	Scoreboard	5	Scoreboard should hold all valuable informations of game, e.g. Player names, score, found words, (…)	
	Dictionary	5	Object which should have access to the whole wordlist, and have a method which checks if given word exists in the dictionary	Yilmaz

Game Logic	Game Flow	5	Game class which can interact with its players, provides methods to interact with itself, controls turn assignments, evaluates score, checks overtime	Yilmaz
	Score Evaluation	5	Method should be able to evaluate the score of placements in the last turn according to the rulebook	
	Placement Check	5	Method should be able to check all placements on the board (Syntax -> w/o checking if formed words exist) according to the rulebook	Yilmaz
Extras	Player Profile (XML)	5	Parse list of player profiles the user has to an XML file, and vice versa	Nicolas
	XML Parser (BoardState, PlayerInfo)	5	Parse information about state of board to XML-formatted String, and vice versa. The same with player information (to send PlayerProfiles to the server)	Nicolas
	Import Dictionary	5	It should be possible to create a Dictionary instance filled with words which are contained in the .txt-file at given path (BinaryTree)	Valentin
	Customize Tile Dist&Score&Dictionary	5	Tile distribution, scores & dictionary should be contained in Game class	Yilmaz
	Sound	2	Provide a static class to play sounds with which are to be saved in a designated directory	Nicolas
	Game History (SQL)	1	Save history of game results which the user participated in	
GUI	Startup Screen	3	View & Controller	Max & Vincent
	Main Menu Screen	5	View & Controller	Max & Vincent
	Settings	1	View & Controller	Max & Vincent
	Manage Player Profiles	5	View & Controller	Max & Vincent
	Play Scrabble Menu	5	View & Controller	
	Create Game	5	View & Controller	Max & Vincent
	Join Game	5	View & Controller	Max & Vincent
	Game Lobby	5	View & Controller	Max & Vincent
	Game	5	View & Controller	Max & Vincent
	Chat	5	View & Controller	Max & Vincent
	Game Result	5	View & Controller	Max & Vincent
	Tutorial	5	View & Controller	Max & Vincent
	Game History	1	View & Controller	Max & Vincent

	Set tile Dist & Score & Dictionary	*5*	View & Controller Passed to game instance when game is started	Max & Vincent
Documentation	Fully-Dressed Use Cases	*5*		Team
	System Sequence Diagrams	*5*		Team
	Operation Contracts	*5*		Team
	Project Plan	*5*		Yilmaz
	Presentation Slides	*5*		
	Description Libraries	*5*		Yasin
	Progress Report	*5*		Team, Vincent
	Preliminary Architecture Diagram	*5*		Valentin
	Mock-Ups	*5*		Max & Vincent
	Domain Model	*5*		Yilmaz

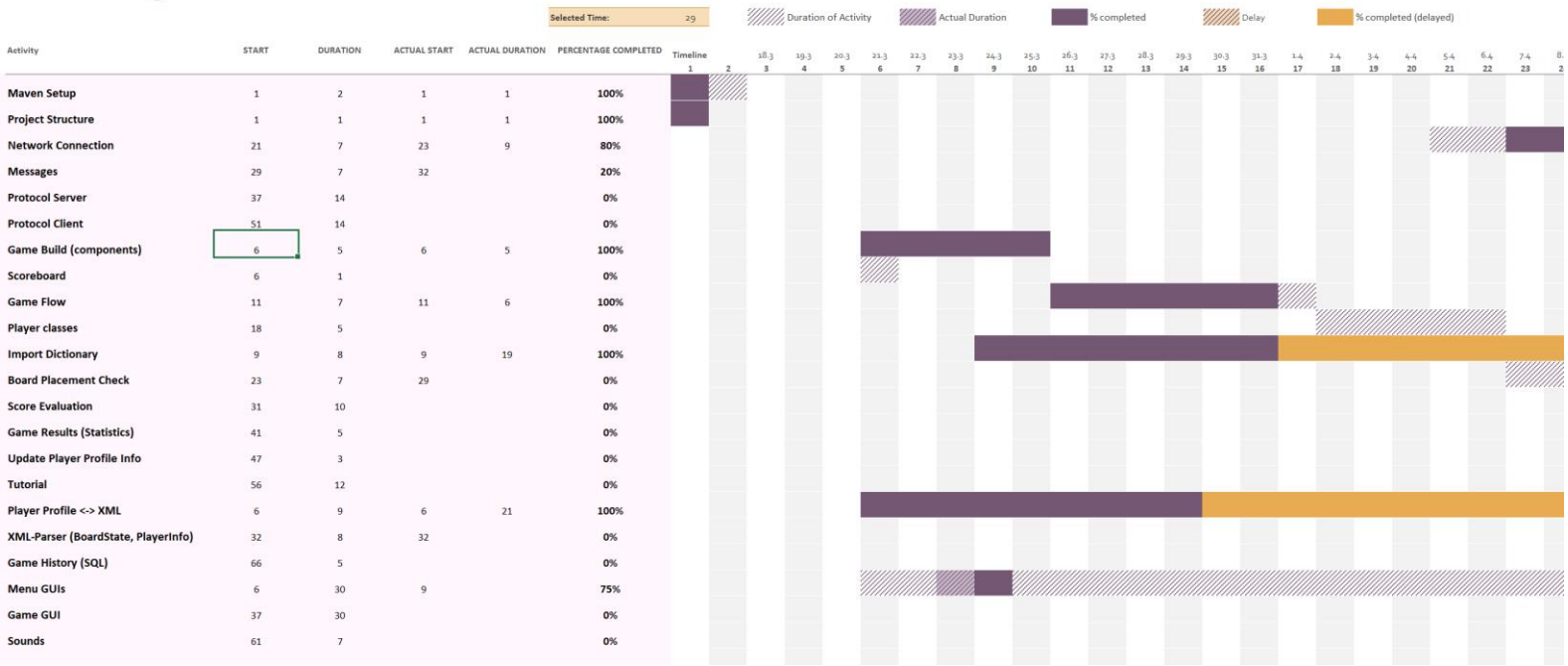
Timeline

Author: Yilmaz Uzun

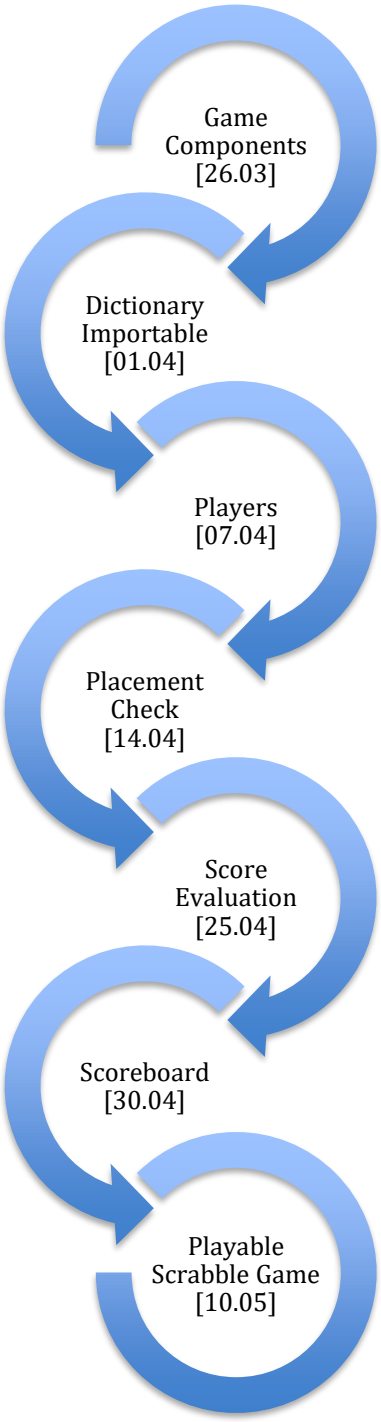
Tasks	Start Plan Date	Plan Duration	End Date	Actual Start Date	Actual Duration	Actual End Date	Completion	Delay	Assigned
Maven Setup	16.3	2	18.3	16.3	1	17.3	100%		Yilmaz
Project Structure	18.3	1	19.3	16.3	1	17.3	100%		Yilmaz
Network Connection	5.4	7	12.4	7.4	9	16.4	80%	4	Yasin, Valentin
Messages	13.4	7	20.4	16.4			20%		Yasin, Valentin
Protocol Server	21.4	14	5.5				0%		Yasin, Valentin
Protocol Client	5.5	14	19.5				0%		Yasin, Valentin
Game Build (components)	21.3	5	26.3	21.3	5	26.3	100%		Yilmaz
Scoreboard	21.3	1	22.3						
Game Flow	26.3	7	2.4	26.3	6	1.4	100%		Yilmaz
Player classes	2.4	5	7.4						
Import Dictionary	24.3	8	1.4	24.3	19	12.4	100%	11	Valentin
Board Placement Check	7.4	7	14.4	13.4					Yilmaz
Score Evaluation	15.4	10	25.4						
Game Results (Statistics)	25.4	5	30.4						
Update Player Profile Info	1.5	3	4.5						
Tutorial	10.5	12	22.5						
Player Profile <-> XML	21.3	9	30.3	21.3	21	11.4	100%	12	Nico
XML-Parser (BoardState, PlayerInfo)	16.4	8	24.4	16.4					Nico
Game History (SQL)	20.5	5	25.5						
Menu GUIs	21.3	30	20.4	24.3			75%		Max, Vincent
Game GUI	21.4	30	21.5						
Sounds	15.5	7	22.5						

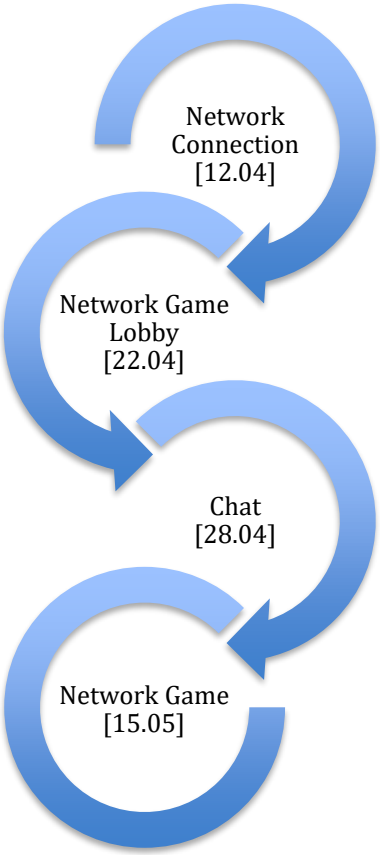
	In Development
	Critical
	implemeneted, pending tests
	done (incl. Tests)

Scrabble 13



The .xlsx workbook with the planning is in the included in the submission as ScrabblePlan.xlsx


Game Components [26.03]
Dictionary Importable [01.04]
Players [07.04]
Placement Check [14.04]
Score Evaluation [25.04]
Scoreboard [30.04]
Playable Scrabble Game [10.05]


Network Connection [12.04]
Network Game Lobby [22.04]
Chat [28.04]
Network Game [15.05]

Short Written Report of Progress

Author: Vincent Hofmann

We approached the development of our application with the Agile Kanban methodology. For that we divided the problem into small-packaged user stories and established dependencies between different 'packages'. Packages are assigned among our team members who are not currently working on any features, preferably to members who are responsible for similar features which were integrated before.

Max and Vincent created the Mockups which help design the GUI, based on which they already wrote some FXML files. In more detail, the Mockups "Create Game", "Game Results Chat", "Game Settings", "Game View", "Join Game View", "Player Lobby View", "Player Profile View", "Player Scrabble View", "Setting", "Welcome View" were created so far. The associated Controllers were coded as well, so first responsive navigation was made possible.

Nico finished the XML-Player-Profile documents successfully, therefore enabling the parsing of XML-files.

Valentin and Yasin established the basic Client-Server network connection, in which they created different classes for the message types that will be sent between clients and server. Also the server-side and client-side protocols that should handle these various message types have been taken care of. Finally the server setup, which is realised with sockets was finished. Until now they are able to connect and disconnect as a Client from a Server. Additionally, Valentin created the classes related to the dictionary to load and store words from a wordlist and change the used wordlist as well.

Next, we will focus more on navigation and the development of the missing GUIs, thus adding more features described by the Mockups. Further, the values which are laying on the scrabble board should be saved and additionally used in order to update the XML files with the current game status. Concerning the network, the goal is to enable sending and receiving messages between the several parts of the game. For instance indicating that a player is ready or a tile has been placed.