Assignment 2

Alex Geenen, Bart de Jonge, Mark van de Ruit, Menno Oudshoorn, Stefan de Vringer

Software Engineering Methods, Group 1

LAN Multiplayer Requirements Document

Below we show the requirements document we created before desigining and implementing LAN multiplayer into our game. This was our assignment of choice (and what our TA had in mind) for this week.

Functional requirements

Must haves

- Ability for two computers to connect over a Local Area Network and exchange data
- Ability to connect two computers via IP-address
- Ability to play a game of Bubble Trouble with two computers in multiplayer, in sync

Should haves

- Low-latency connection over Local Area Network
- Ability to set player name before game
- Player name should be shown above players in game

Could haves

- Ability to connect without IP-address
- Ability to display Ping

Won’t haves

- Ability for more than two players to play a game together

Non-functional requirements

- Multiplayer LAN is OS-agnostic
- Minimum version of Java required is 1.8

Responsibility driven design

To work from a responsibility-driven design perspective, we created some extra CRC cards, and updated old ones where necessary, to help with the design process.

NEW CLASSES

Class Name: Client	
Superclass: none	
Subclasses: none	
Responsibilities	**Collaborators**
Communicate with host	Host
Update data in game	GameState
Receive info about changes from host	GameState
Send info about changes to host	GameState

Class Name: Host	
Superclass: none	
Subclasses: none	
Responsibilities	**Collaborators**
Communicate with client	Client
Update data in game	GameState
Receive info about changes from client	GameState
Send info about changes to client	GameState

Class Name: MenuMultiplayerState	
Superclass: BasicGameState	
Subclasses: none	
Responsibilities	**Collaborators**
Allow player to join a LAN game	Button, Client
Return to the starting screen	Button, MenuMainState
Allow player to host a LAN game	Button, Host
Allow player to enter their player name	Textfield
Allow player to enter IP to join LAN game	Textfield

Class Name: Separator	
Superclass: none	
Subclasses: none	
Responsibilities	**Collaborators**
Be a seperator between different parts of a menu	MenuMultiplayerState, MenuSettingsState, MenuMainState, MenuGameoverState
Appear to be a separator by drawing text	RND

Class Name: Textfield	
Superclass: none	
Subclasses: none	
Responsibilities	**Collaborators**
Be a field that you can use to enter text, in menus	MenuGameoverState, MenuSettingsState, MenuMainState, MenuMultiplayerState

Class Name: Popup	
Superclass: none	
Subclasses: none	
Responsibilities	**Collaborator**
Show a warning to the user	MenuGameoverState, MenuSettingsState, MenuMainState, MenuMultiplayerState, Separator, RND
Allow user to respond	Button, RND
Appear on screen as a popup	Separator, RND

UPDATED RESPONSIBILITIES

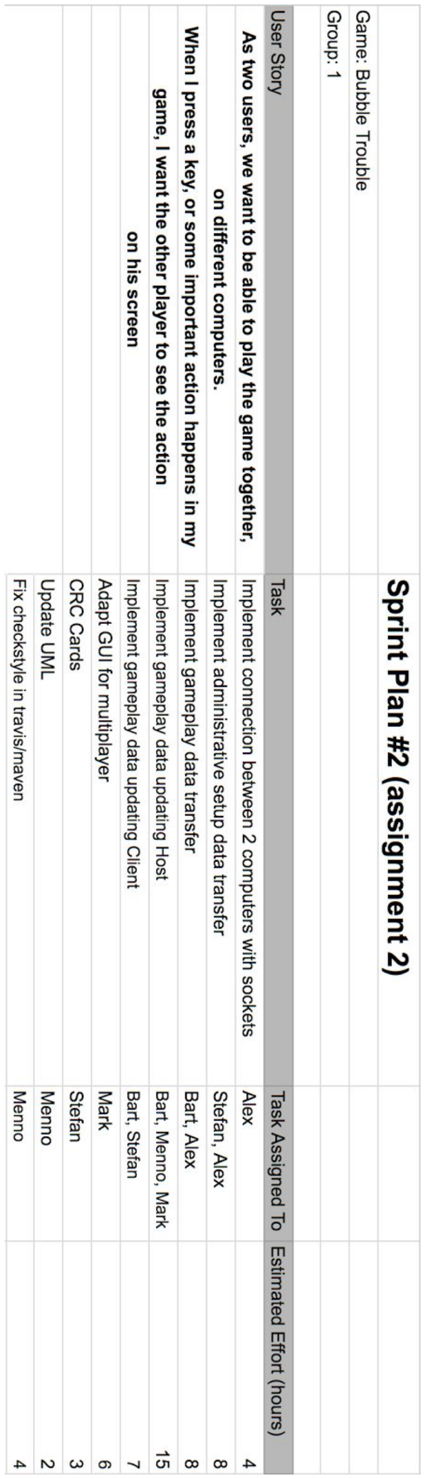
Class Name: GameState	
Superclasses: none	
Subclasses: none	
Responsibilities	**Collaborators**
Manage collisions	Player, Gate, Rectangle, Bubble, Weapon
Keep track of lives	Lives
Keep track of player death, switch to necessary state	GameOverState, Lives
Spawn powerups	Powerup
Show pause menu when game is paused	PauseState
Keep track of time left	Countdown
Keep track of points/coins	Points, Coin
Keep track of levels	Level, LevelContainer
Send data to other player in LAN game	Client, Host

Class Diagram

Of course, to help with our design, we also created a class diagram using UML. **Due to the ridiculously preposterous 5K size of the image**, it can only be viewed separately in the same folder this document is located in. It would not be able to fit inside of a document.

Sprintplan

Below we show our sprintplan for this sprint. It is also contained separately in the same folder this document is located in.



Sprint Plan #2 (assignment 2)

Game: Bubble Trouble

Group: 1

User Story	Task	Task Assigned To	Estimated Effort (hours)
As two users, we want to be able to play the game together, on different computers.	Implement connection between 2 computers with sockets	Alex	4
	Implement administrative setup data transfer	Stefan, Alex	8
When I press a key, or some important action happens in my game, I want the other player to see the action on his screen	Implement gameplay data transfer	Bart, Alex	8
	Implement gameplay data updating Host	Bart, Menno, Mark	15
	Implement gameplay data updating Client	Bart, Stefan	7
	Adapt GUI for multiplayer	Mark	6
	CRC Cards	Stefan	3
	Update UML	Menno	2
	Fix checkstyle in travis/maven	Menno	4