Assignment 1

Team number: 35
Team members

Name	Student Nr.	Email
Elif Kalkan	2691928	e.kalkan@student.vu.nl
Jay Doerga	2696009	j.a.doerga@student.vu.nl
Jazzley Louisville	2685898	j.j.j.louisville@student.vu.nl
Zefan Morsen	2691045	z.b.morsen@student.vu.nl

Introduction

Author(s): Jay & Jazz

The aim of this project is to implement a short-and-simple API/version of a control room loosely based on the massively popular Netflix show 'Squid Game'. Our control room brings the enticing aspect of player elimination and master control,i.e pausing/restarting/sequencing games in a specific order, to different games. We have decided as a team to implement some of the following features that would allow such control:

- Option to eliminate a player
- Option to start games in a certain sequence
- The ability to have an overview of the current games being played as a master.
- The ability to invite players
- The ability to influence the game as a master(by pausing/restarting the game, messaging players, etc).

Summary of how the control room/api is used and its stakeholders:

- 1 master who has the ability to influence the game in whatever way they see fit
- Unlimited amount of users on invite basis issued by the master
- Developers are able to use the API and add it to their games, to allow master control. On a plug-and-play basis.
- Relevant data is sent from the game towards the server and back to the master client.
- The game status updates in real-time, thus allowing influence from the master console
- The users are able to play games when invited to a certain host.

The customizations available to developers will be as follows:

- Receiving JSON data, implementing this received data into their own game based on the accompanying documentation provided, this data is a request sent by the master to trigger a specific game state instance.
- Send JSON data, formatted to the specification given in the accompanying documentation provided, this data is an acknowledgment sent back by the developer to confirm the request by the master has been triggered.

Main system modules:

- UI: This will be the UI the master interacts with when issuing commands/triggering specific game states. There will be graphical components in the UI to ease the use of the system.
- Parsing module: This is the middleware used to parse the JSON data to java objects
- Server: Data is sent back and forth between the master console, the server, and the game platforms.

The following is a description of how the control room works:

- Master opens the application
- Chooses a sequence of games to be played
- Invites specific players to play the game
- Observes the games, with data about the current game state being sent back in real-time.
- Has the ability to pause the game, remove players from the game or end the game early, etc.
- If a master triggers an event, the data is sent to the server and from the server to the game that is currently being played.
- The control ends when the master triggers a self-destruct event.

Features

Author(s): Zefan

Functional features

The operator of the control room is the game master. Therefore, the functional features are based on the actions the master is able to perform.

ID	Short name	Description	Champion
F1	Start game	The master should be able to start the game that the users will play.	Jazzley
F2	Pause/Resume	The master should be able to pause the current game played and resume it at any point.	Jazzley
F3	Games	The master should be able to see all the games that are imported to the program.	Jay
F4	Sequence	The master should be able to create a list of games to play in an order of his choosing.	Elif
F5	Users	The master should be able to invite the users to the game that will be played.	Jay
F6	Status	The master shall be able to see a list of the players that are still alive.	Zefan
F7	End	The master should be able to end the game at any time.	Zefan

Quality requirements

Author(s): Elif

The quality requirement were made based on the following statements:

- The master must see the updates of a player immediately
- The events triggered by the master must be sent immediately
- The adding and removing of a game by a developer should be easy and fast
- Only the master should be able to start and pause the game so it's fair
- The GUI of the control room should be easy to understand and controllable by the master

ID	Short name	Quality attribute	Description
QR1	Instantaneous results	Performance	If a user's state changes, the updates should be sent to the master in 0.2 seconds.
QR2	Continuity	Reliability	The current game state should remain until an event is triggered by the master.
QR3	Appearance	Consistency	Pause, play, and stop buttons of the control room shall be the same size
QR4	Speed	Performance	Once the master starts, pauses, or stops the game, the result should be available within 0.2 seconds
QR5	Modifiability	Maintainabailty	A developer should add new games or remove old games without disturbing an ongoing game.
QR6	Security	Control	Only the master should start, pause or stop the games
QR7	GUI	Appearance	The game should be user-friendly and intuitive

Java libraries

Author(s): Jay & Jazz

JFoenix

Used for styling the user interface of the system. We chose it among others because...

Fastjson

We will use it for reading and writing JSON configuration files containing the description of the cards of the deck. We chose it because it is easy to configure and use from Java code and preliminary experiments make us confident about its correct functioning...

JavaFX
Used for the front-end styling of our application. We chose JavaFX because it's open-source, efficient, and has a fully-featured toolkit for developing rich client applications.

Apache HTTP Client
Used for HTTP calls to get information about players and lobbies from our server.

Time logs

Team Number	35		
Member	Activity	Week number	Hours
Jazzley & Jay	Introduction/project description	1 & 2	2
Jazzley & Jay	Listing libraries	1	2
Zefan		2	2
Elif		2	2
All of us	Time logging	1 & 2	0.5
All of us	Mentor meetings	1 & 2	1.5
		Total	10