

Current projects: [Splash!](#) 

[Blue Noise Splash](#)  

[Pfad mit Lichtungen \(Path with glades\)](#)

[Splash Noise](#)

[Splash Noise Blues](#)

hypersound setup

WORK IN PROGRESS

Created: 10.07.2022 10:58

Last Change: 10.09.2022 09:15

Autoren: Friedhelm Hartmann, Heinz-Josef Florian

Table of Contents

TABLE OF CONTENTS	1
1 COMMUNICATION SETUP	2
1.1 HARDWARE	2
1.2 SOFTWARE	2
1.3 TEAMVIEWER SETUP	2
1.3.1 <i>For the Sound Agent (the remote player person)</i>	<i>2</i>
1.3.2 <i>For the Sound Host (the onside performer person)</i>	<i>4</i>
1.4 CONNECTING VIA TEAMVIEWER	6
1.4.1 <i>For the Sound Agent</i>	<i>6</i>
1.4.2 <i>For the Sound Host.....</i>	<i>7</i>
2 SESSION MANAGEMENT	8
2.1 GENERAL COMMUNICATION	8
2.2 AVOIDING CONTROL CONFLICTS	8
2.3 SCREEN CONTROL FOR SIMULTANEOUSLY PLAYING	10
2.4 AUDIO STREAMING FROM THE HOST COMPUTER	10
2.5 FILE TRANSFER	11
2.6 MOBILE USERS	11
2.7 ENDING A SESSION.....	13
3 PARALLEL AUDIO AND VIDEO COMMUNICATION	14
3.1 GENERAL	14
3.2 PARALLEL SKYPE SESSION.....	14
3.3 MEETING PROCEDURE	14
4 SESSION RECORDING WITH COREL MULTICAM.....	16
4.1 GENERAL	16
4.2 SETTINGS	16
4.3 START AND STOP RECORDING.....	18
5 SCREENPLAYER.....	19
5.1 BASICS	19
5.2 PLAY FUNCTIONS	20
5.3 CONFIGURATION.....	23
5.3.1 <i>Files and folders.....</i>	<i>23</i>
5.3.2 <i>Sound file configuration in samples.js</i>	<i>23</i>
5.3.3 <i>Touchp(o)int Setup in samples.js.....</i>	<i>25</i>
5.3.4 <i>Score in samples.js.....</i>	<i>26</i>
5.3.5 <i>Screenplayer Customization in Screenplayer.htm</i>	<i>28</i>
6 PLAYING EXAMPLES	29
6.1 INTRODUCTION	29
6.2 PLAYING RULES EXAMPLE 1: „BLUE NOISE SPLASH“	29
6.3 PLAYING RULES EXAMPLE 2: “SPLASH NOISE”	30
6.4 PLAYING RULES EXAMPLE 3: “SPLASH NOISE BLUES”	30
6.5 PLAYING RULES EXAMPLE 4: “PATH WITH GLADES”	31

1 Communication Setup

1.1 Hardware

- ◆ Laptop, preferable Windows PC, MAC should work
- ◆ Headphones connected to stereo headphone plug via cable (Bluetooth problematic) or computer loudspeaker
- ◆ Sensitive mouse or touchscreen

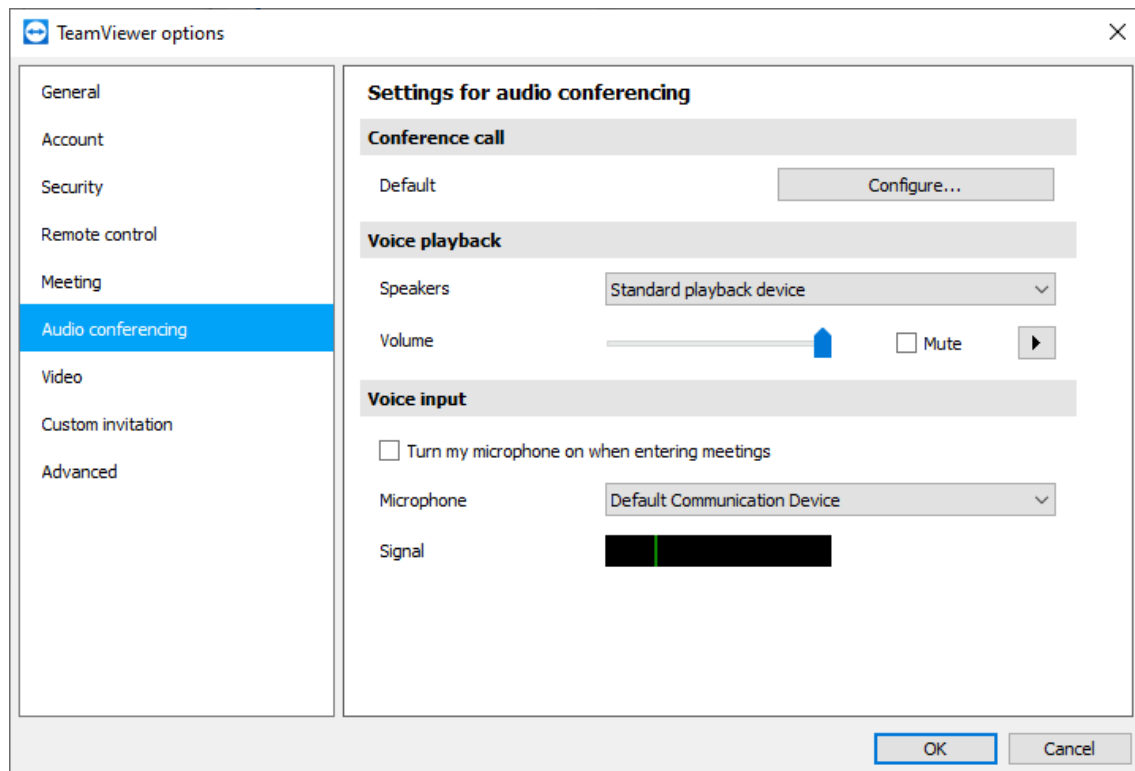
1.2 Software

- ◆ TeamViewer private installation (free of charge), newest version recommend <https://www.teamviewer.com/en/teamviewer-automatic-download/>
- ◆ Skype, private version, see <https://www.skype.com/en/>

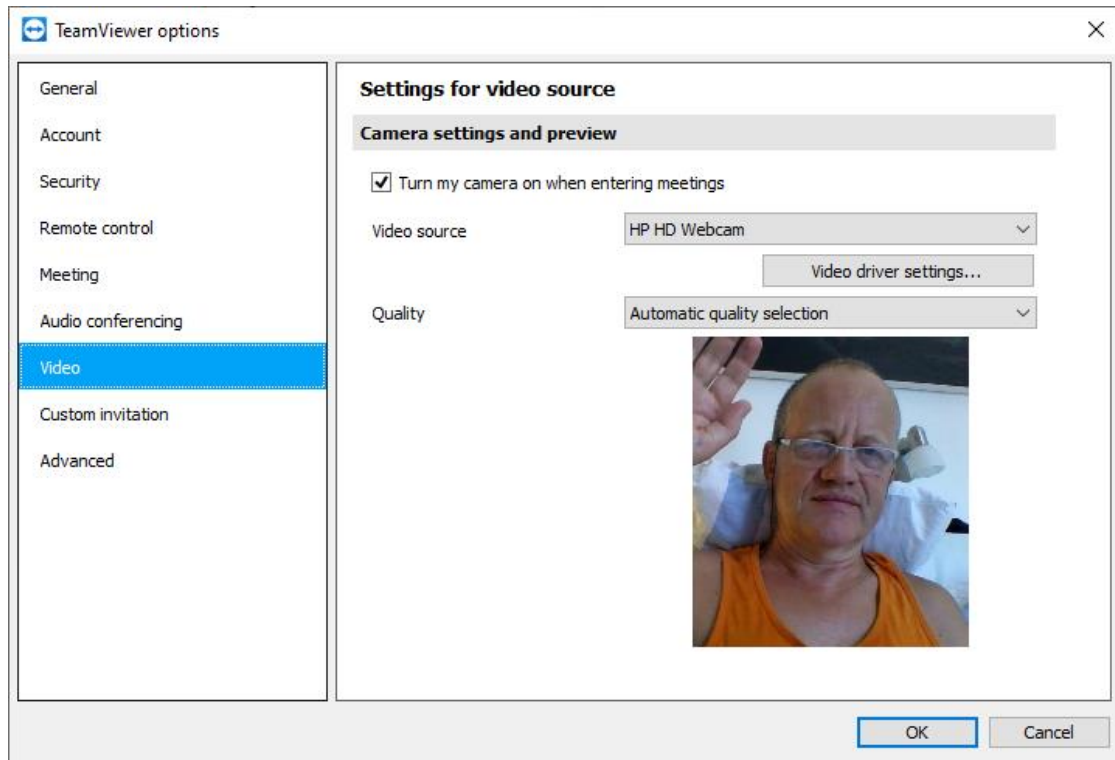
1.3 TeamViewer Setup

1.3.1 For the Sound Agent (the remote player person)

1. After Download, click `TeamViewer_Setup_x64.exe` to install.
2. Launch the program.
3. Click **Options** (tooth wheel)
4. Click **Audio conferencing**
5. Check the following settings:

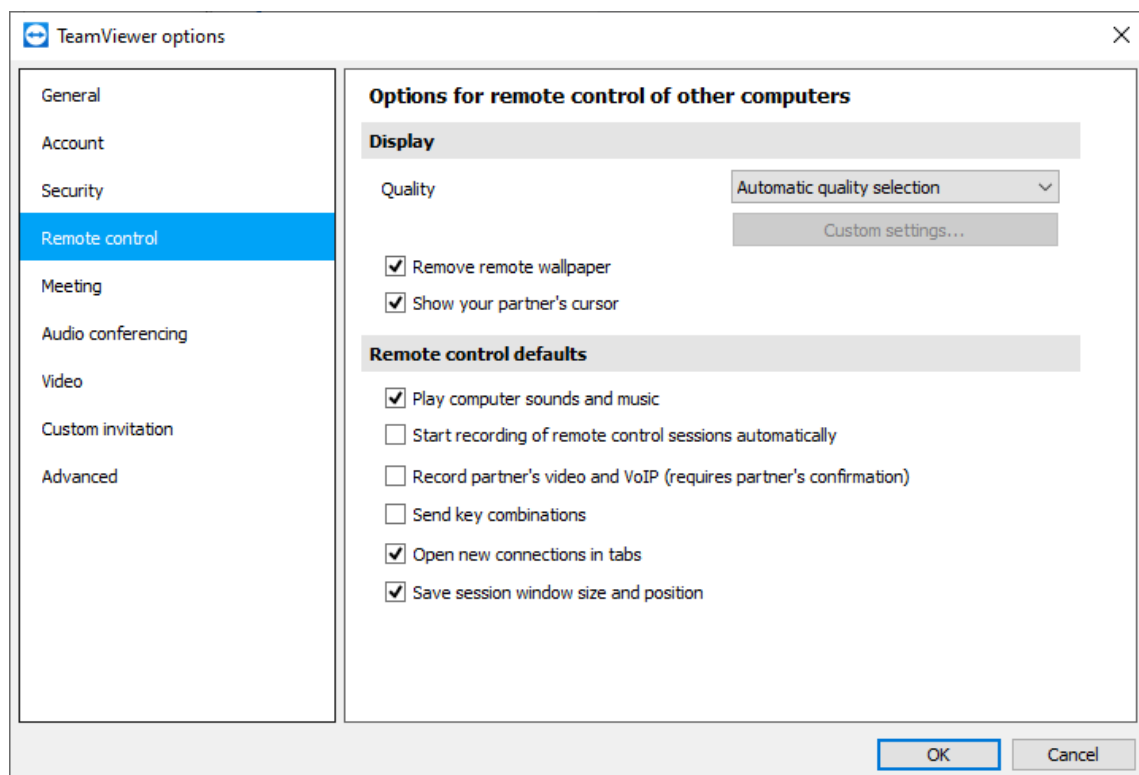


6. Click **Video**
7. Check the following settings:



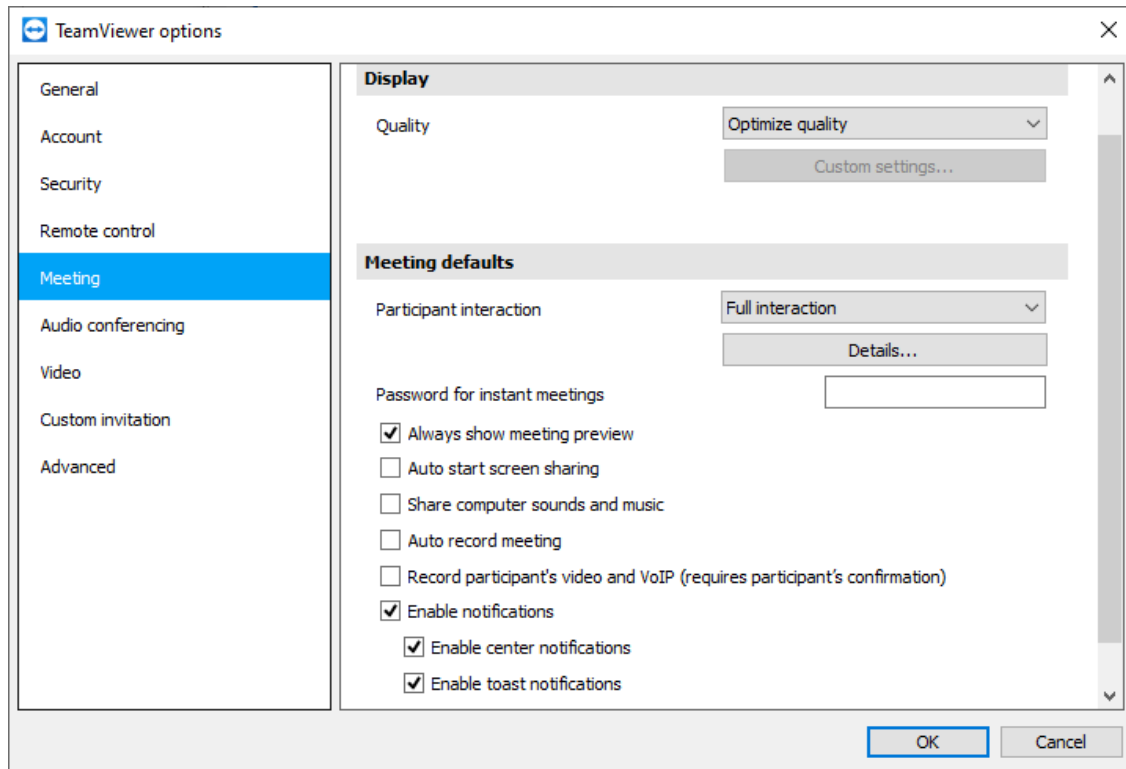
You need to see yourself in the picture. Did you select the correct **Video source**?

8. Click **Remote control**
9. Check the following settings:



10. Click **Meeting**

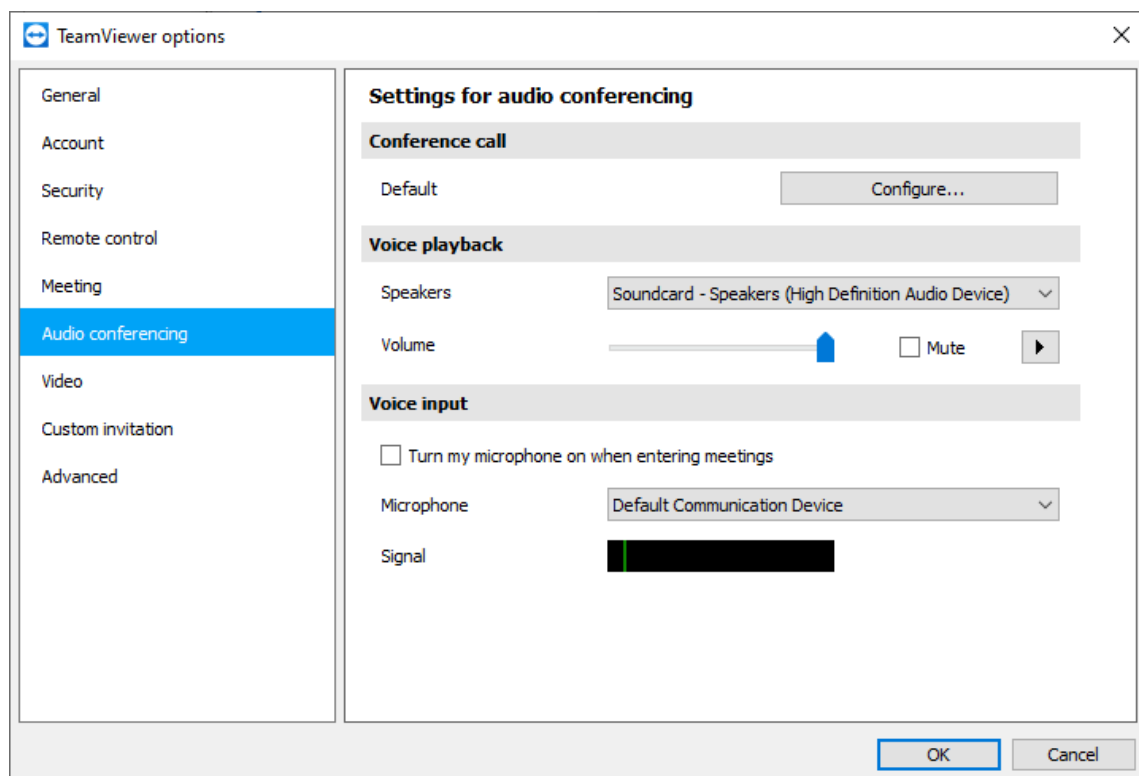
11. Check the following settings:



1.3.2 For the Sound Host (the onside performer person)

1. Follow 1.3.1. steps 1.-4.

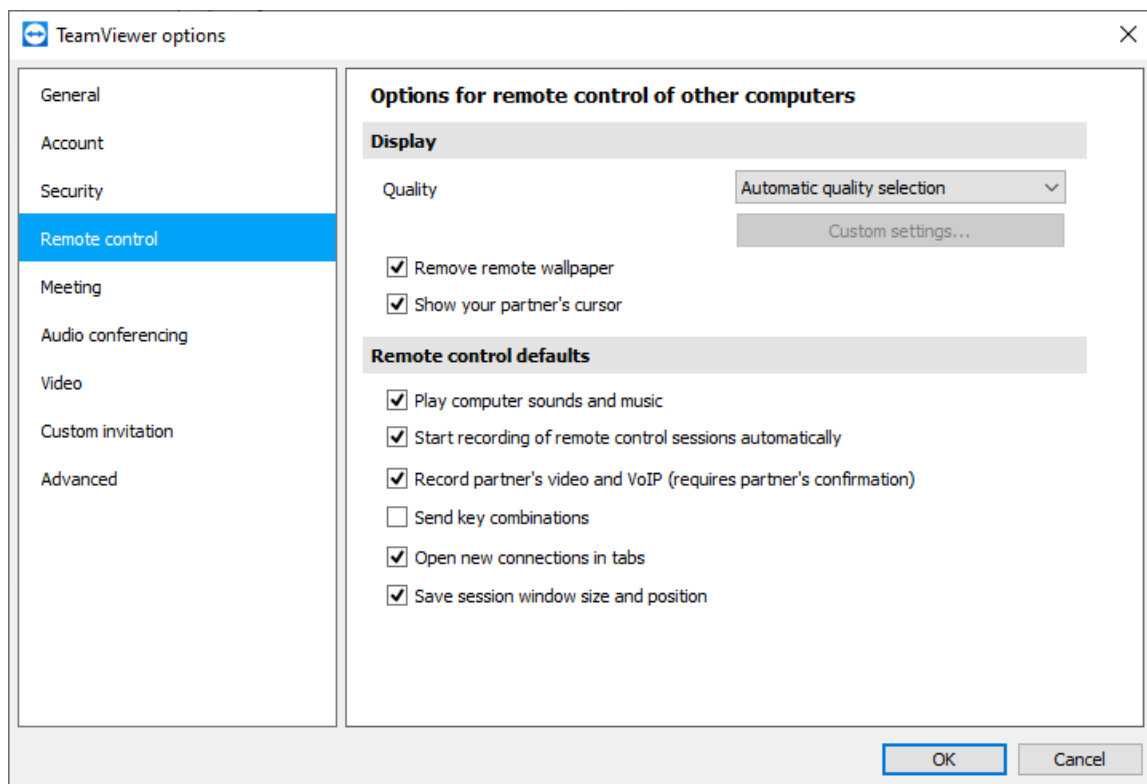
2. Check the following settings:



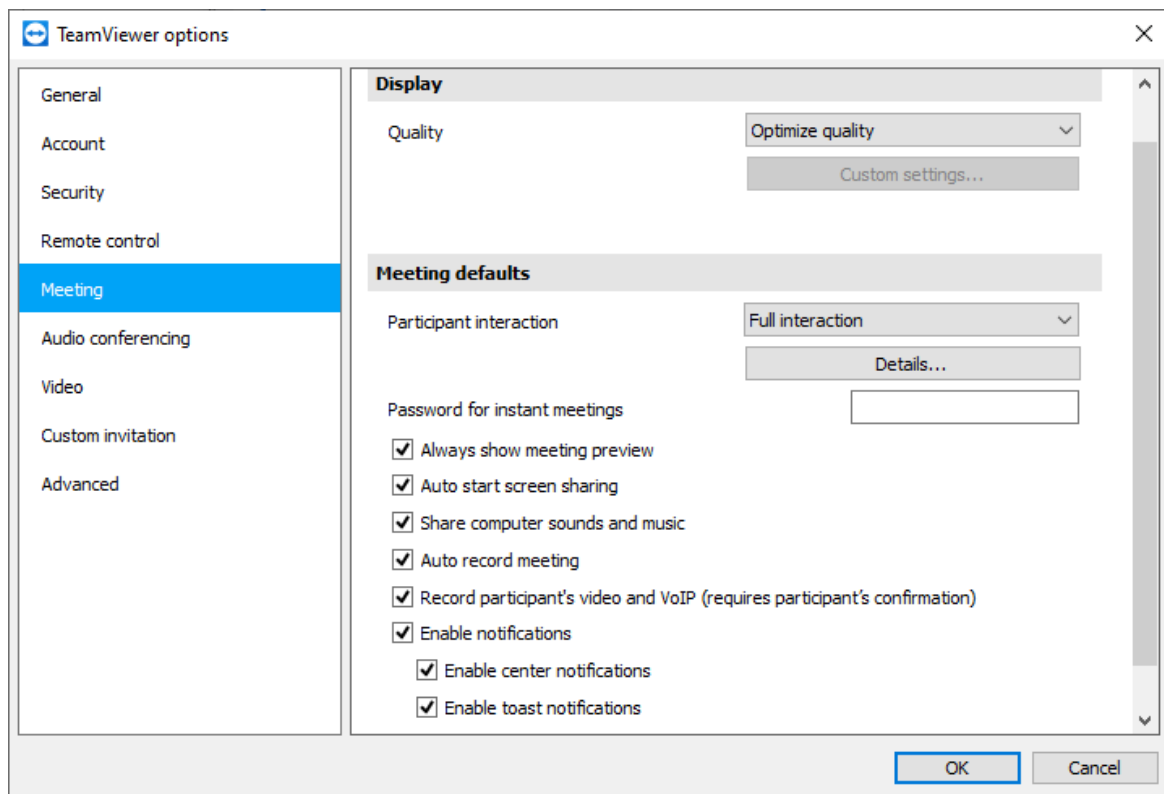


When performing a parallel communication meeting like the recommended Skype application (see 3. Parallel Audio and Video Communication), the above **Mute** option need to be temporarily checked, in order to avoid echo effects for the Sound Agents. At the same time, TeamViewer will NOT send any of the sound information from the host computer.

3. Follow video settings above (step 6. and 7.)
4. Click **Remote control**
5. Check the following settings:



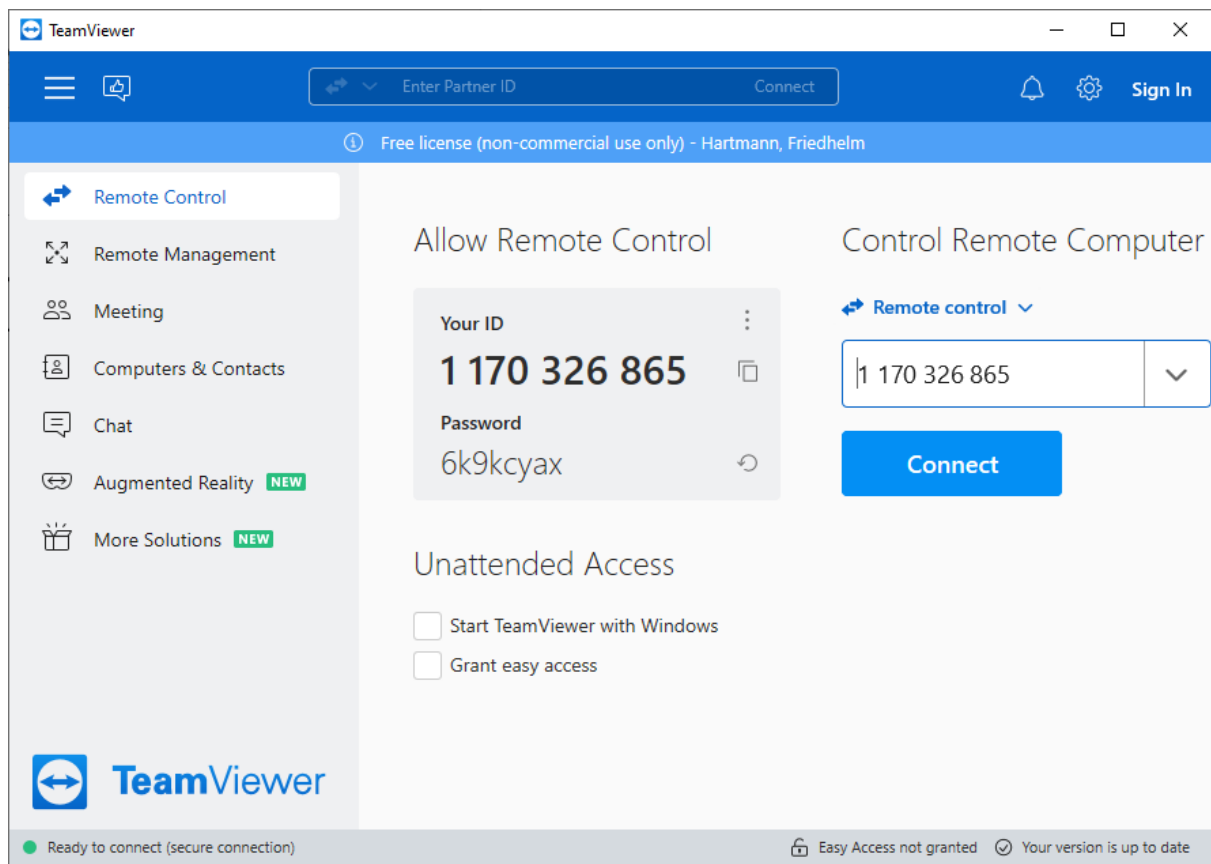
6. Click **Meeting**
7. Check the following settings:



1.4 Connecting via TeamViewer

1.4.1 For the Sound Agent

1. Open TeamViewer
2. Enter **1 170 326 865** in the **Remote control** field:



3. Click **Connect**
4. Get Password from Sound Host and enter it into the Password dialog and click **Log On**. The Host computer screen will be shown and can be already controlled.



At this stage, please do not use your mouse at the Host computer screen. This screen may be minimized for your own computer screen and should be maximized for the musical performance.

1.4.2 For the Sound Host

1. Open TeamViewer
2. Tell the Sound Agents to put Your ID into the **Control Remote Computer** field and to press **Connect**.
3. Tell the Sound Agents the password. After they enter the password, the connection is immediately enabled and the Sound Agents can view and control the Host screen.



At this stage, no one should operate the Host screen except the Sound Host.

2 Session management

2.1 General communication

TeamViewer Remote Control allows multiple communication channels simultaneously, including:

- ◆ Screen control for simultaneously playing – Used for the project
- ◆ Chat – Used with Skype
- ◆ Video display – Used with Skype
- ◆ Audio communication – Used for the project to transfer the played piece from the Host computer to all Sound Agents

Voice conversations used with Skype

- ◆ File transfer – Optional
- ◆ Use of mobile phones – Offers the option to involve the audience at the Host presentation site (users need to download "Remote control" app from TeamViewer)

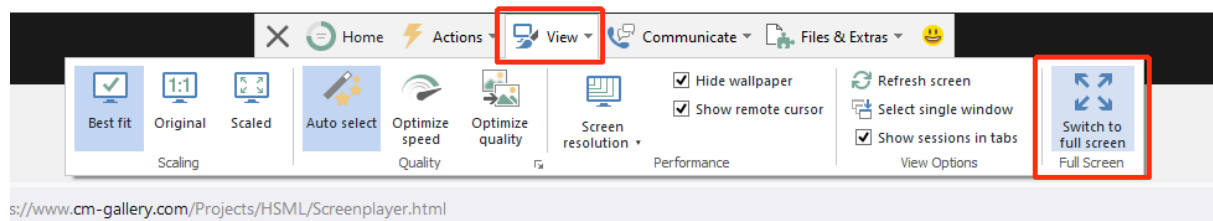
2.2 Avoiding control conflicts

As there are no automatic control restrictions, following rules should be followed:

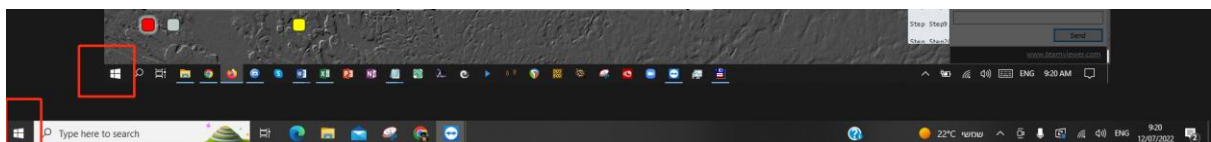
1. Sound Agents should never confuse the Host screen with their own computer, especially when operating their TeamViewer control panel.

To do so,

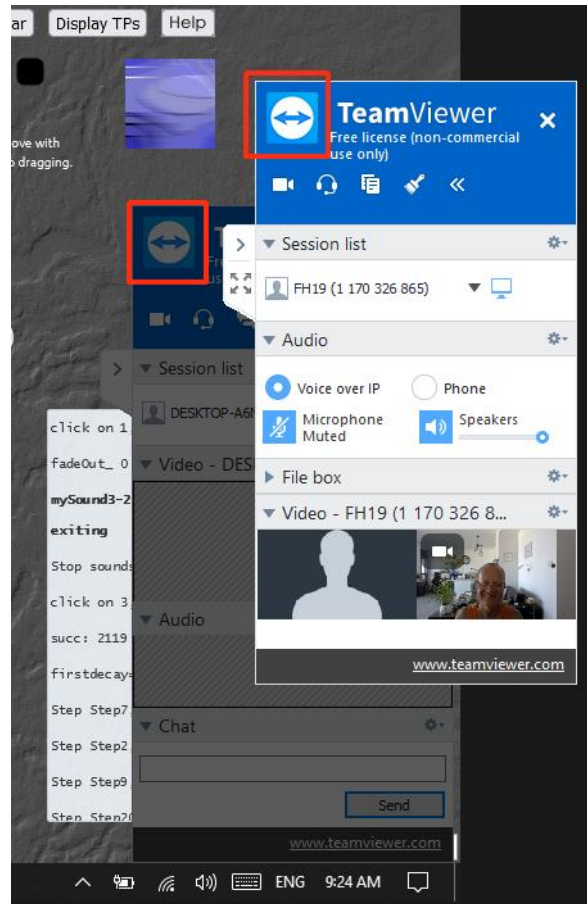
- ◆ TeamViewer should NOT be used in Full screen mode. For this to happen, the **View** menu should show the following **Full Screen** button:



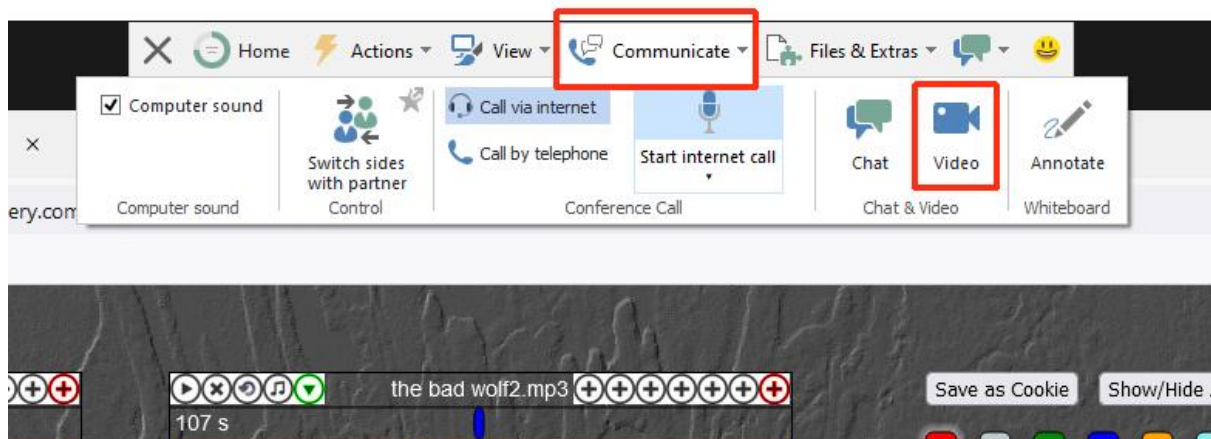
Make sure you as a Sound Agent always see TWO task bars, the lower one being the one of your computer.



- ◆ Always use the LOCAL TeamViewer control panel, which is shown in strong blue and not in shaded blue. The shaded blue panel belongs to the Host computer and should never be operated.



If you can't see your local panel, go to the top control bar, select **Communicate** and click **Video**. The panel display will turn back.



2. Each Sound Agent will only operate on the Screenplayer window.

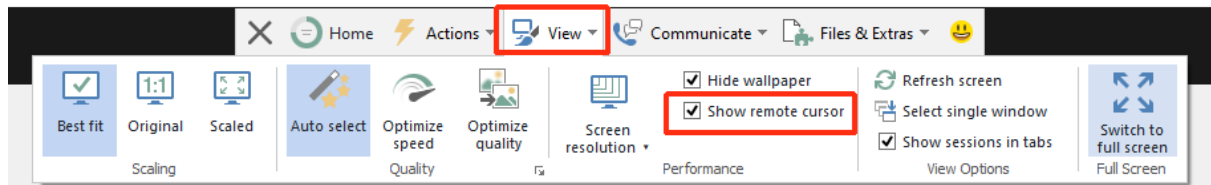


As a general rule, **highlighted** (enlarged) touchp(o)ints of a certain color that is **dedicated to the player** will be used (project pending, see 6.2.).

2.3 Screen control for simultaneously playing

Sound Agents and the Sound Host control the Screenplayer using the mouse or touchscreen simultaneously.

In order to see the other parties operating the one host cursor available, the **Show remote cursor** option in the **View** menu should be enabled (see also chapter 1.3.1 under Remote control):



Specific rules for the musical interaction are project pending and will be shared within the actual working sessions.

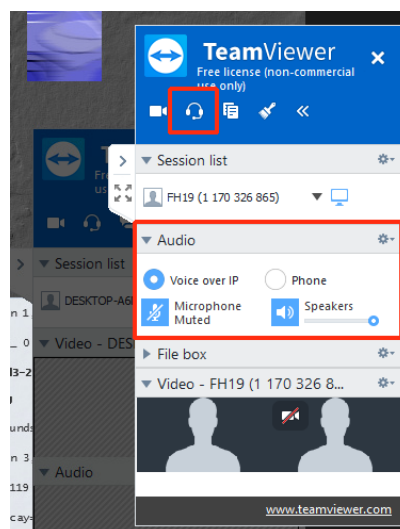
2.4 Audio streaming from the Host computer

The Audio streaming works in high resolution on each of the participating computers and mobile phones (if used). However, due to Internet conditions, a time delay up to one second must be taken into account, which is relevant for the course of the sound interaction between the players. However, no matter how the sound clicks are performed on the Host computer, each of the listeners will get the exact same sound mixing result, may it be delayed or not.

When parallel Voice over IP (for example via Skype) is used for greeting and other communication purposes, the Audio streaming on the Sound Agent's computers will be muted by the Sound Host, while the sound would continue on the Host computer's output, regardless the spoken communication.

Mobile users would get both audio streams as well but are not able to communicate by speech (nor by video).

Sound Agents should mute the TeamViewer audio channel as shown below in order to avoid undesired noise. In order to view the audio control, the headphone button at the TeamViewer control panel need to be used:



2.5 File transfer

TeamViewer also offers a very convenient way to share Files between the computers should this be needed.

2.6 Mobile users

Mobile users are not regarded as usual Sound Agents, but depending on the project, it is possible to engage them as well and to take active part in the musical performance.

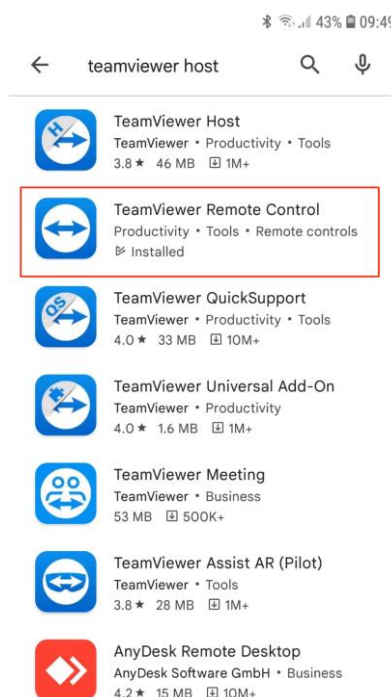
Mobile users cannot share their video nor participate in Voice over IP. However, they are able to control the Host screen as well while being able to resize comfortably their screen to any size required and listening to the high resolution audio stream of the host computer of not present at a live Host presentation of the Screenplayer.



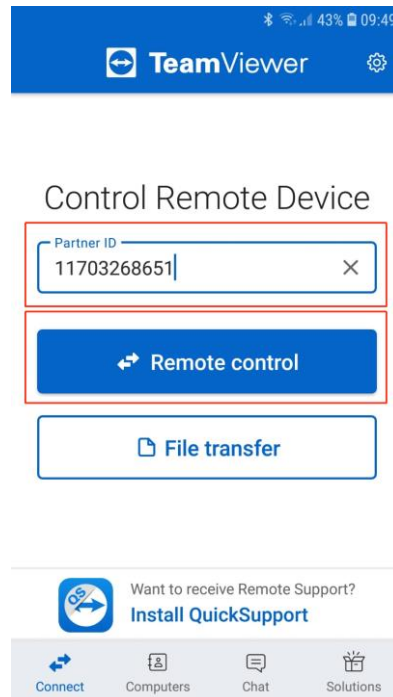
Should the loss of the audio stream be encountered, reconnecting the app usually solves the problem.

In order to allow a Mobile user to connect to the Host's Screenplayer, the following must be done:

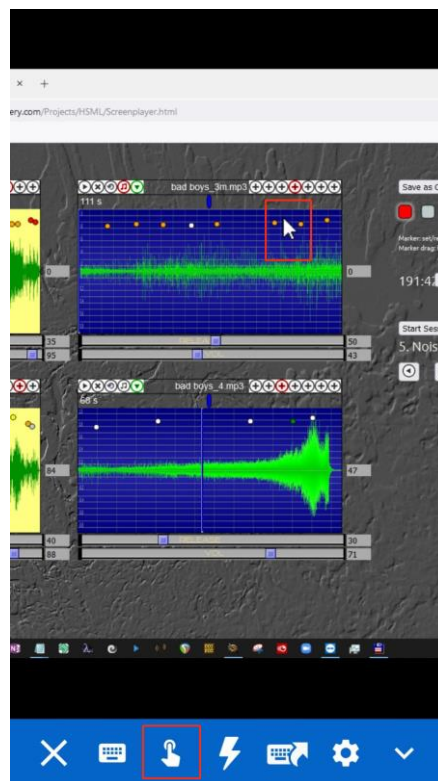
1. Download the **TeamViewer Remote Control** app for free.



2. Open the app and enter the Host's ID, then click **Remote Control**:



3. Enter the Host's **password** and click **OK**. The Host's screen will appear immediately. You may now be able to resize the screen using two of your fingers and select the desired **Interaction mode**, using the Interaction mode button below, choosing between **Mouse** or **Touch mode**.



2.7 Ending a session

Each participant can end his session participation independently and a message will be sent to the host accordingly.

The Sound Host closes the session using the close symbol at the upper right of the TeamViewer control panel.

A TeamViewer session will terminate automatically, if no activity occurred after a certain period of time.

This version of TeamViewer Remote Control can only be used for **private** purposes. Any other use requires acquiring licenses.

3 Parallel Audio and Video communication

3.1 General

TeamViewer RemoteControl does not allow agents communicating with each other, only with the Host.

Therefore it is recommended to use Skype in parallel for session related communication and presenting the acting agents to the audience.

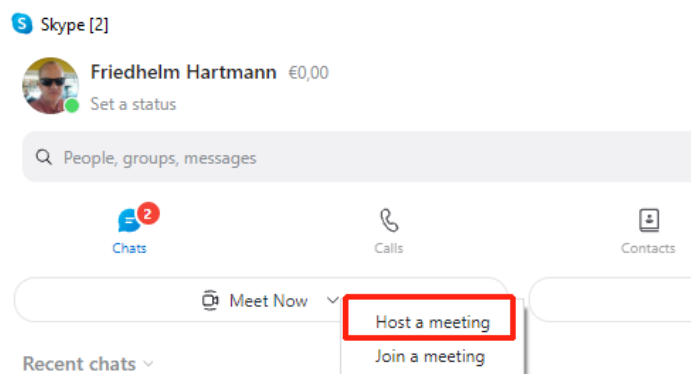
3.2 Parallel Skype session

TeamViewer and Skype audio used to work smoothly in parallel. This means that during a playing session with TeamViewer, the players can still communicate via Skype on top of it.

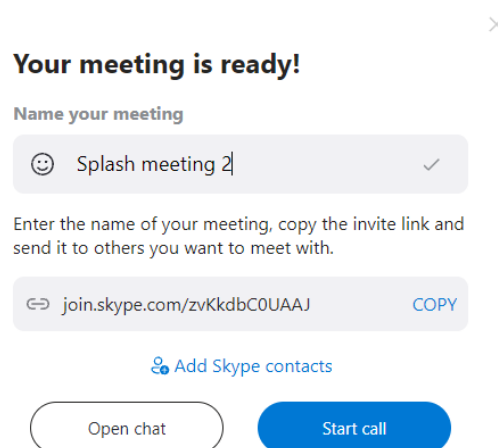
However, during a presentation of the screenplayer, ALL Skype microphones must be set on MUTE.

3.3 Meeting procedure

1. The Sound Host launches Skype and opens a meeting clicking **Meet Now** and selecting **Host a meeting**:



2. The Sound Host copies the meeting link and forwards it to the meeting participants using WhatsApp or alike.



3. The Sound Agents open Skype as above, clicking on **Meet Now**, but selecting **Join a meeting** instead. Then, they enter the meeting link that previously has been provided to them.

4 Session recording with Corel MultiCam

4.1 General

Although TeamViewer has an easy to use screen recording facility, its use is limited:

- ◆ It is only working for a client, not for the host computer.
- ◆ The audio recording showed interruptions during tests. (This may also happen for the second tool suggested below.)
- ◆ The file conversion creates extreme heavy **.avi** video files.

As tool of choice for host computer recording of a hypersound session, **Corel MultiCam Capture Life** can be used both for

1. Complete screen recording with high quality audio.

Recording format is **.mov** with an approx. data volume of 100KB per 15 secs (per following settings; .mp4 would cause the app to freeze).

2. Parallel recording of outer space using the computer's webcam together with additional outer computer microphone sound for use of computer based lecture recording.

Recording format is **.mp4** with an approx. data volume of 10KB per 15 secs (per following settings).

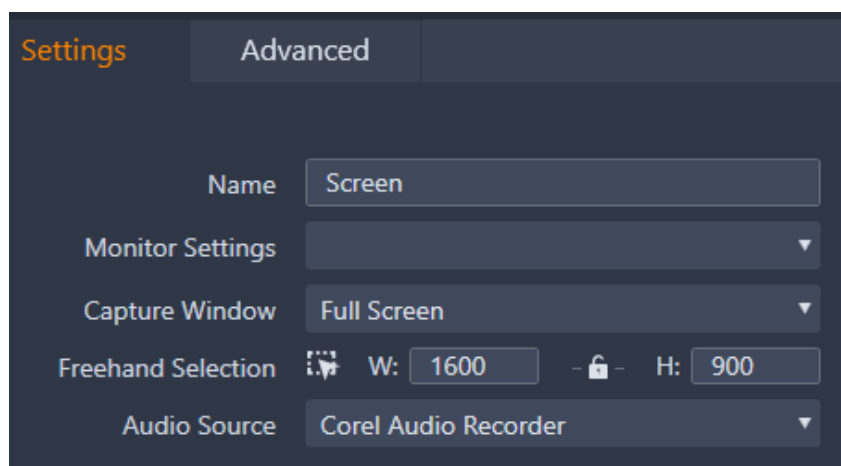


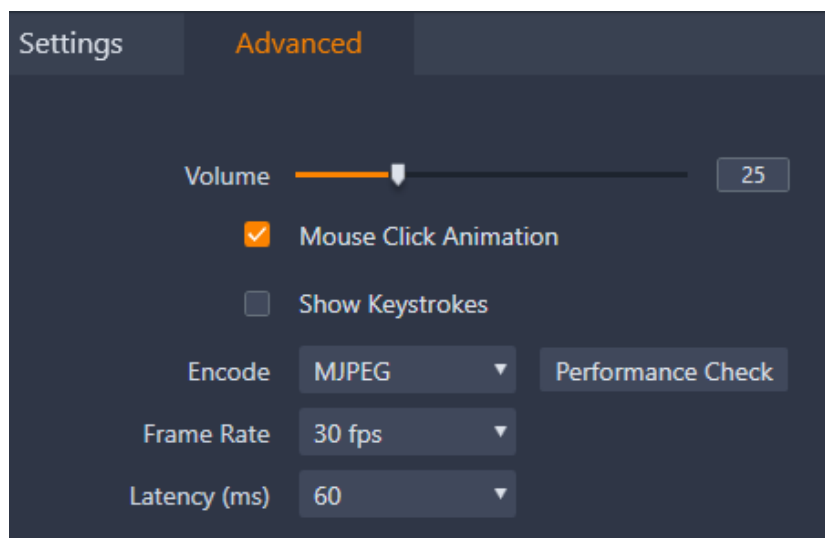
With suitable placement of the host computer, the audience and/or the lecturer can simultaneously be recorded together with the computer screen plus computer audio. **However: Performance issues may occur.**

- ◆ It is strongly recommended NOT to use Bluetooth connection. Use **headphones with cable** via the miniplug stereo output connector of the computer.
- ◆ Also make sure, **no sounds** are used for any app in the host computer (like WhatsApp, etc.).
- ◆ **Refresh** the browser with the Screenplayer before starting and then press F11.

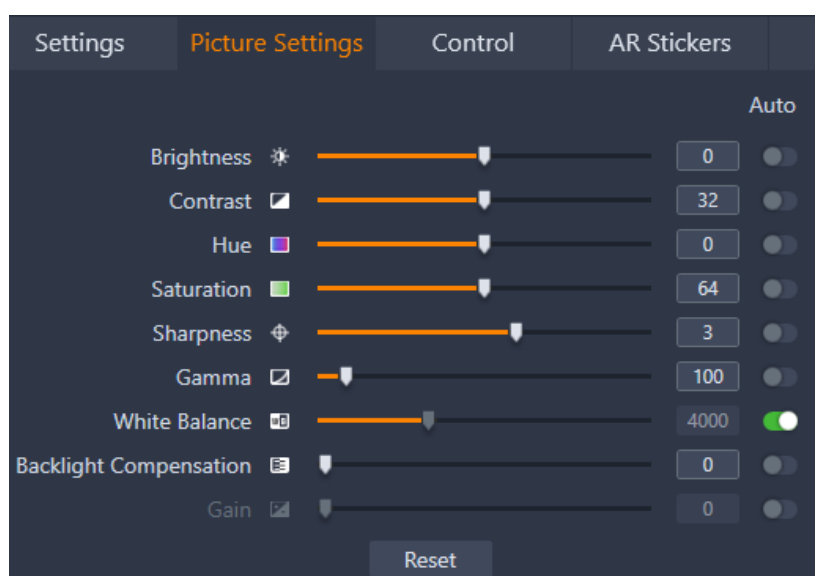
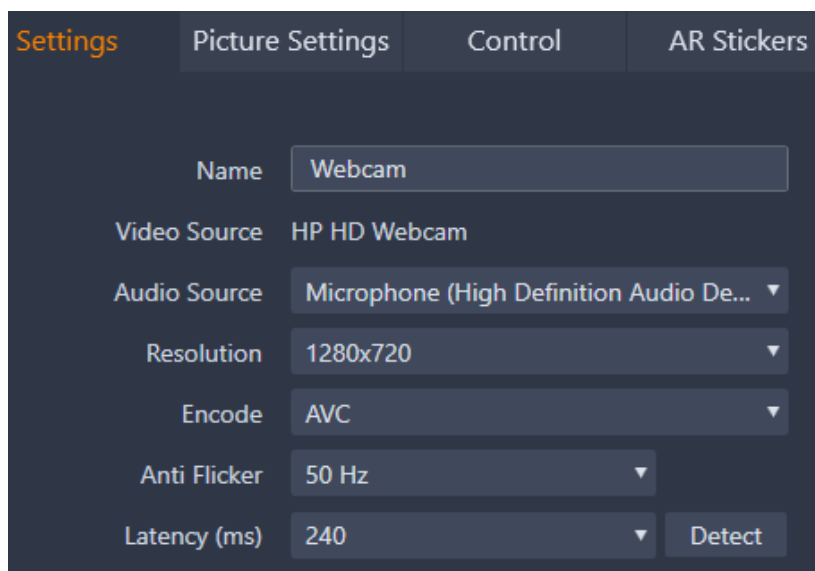
4.2 Settings

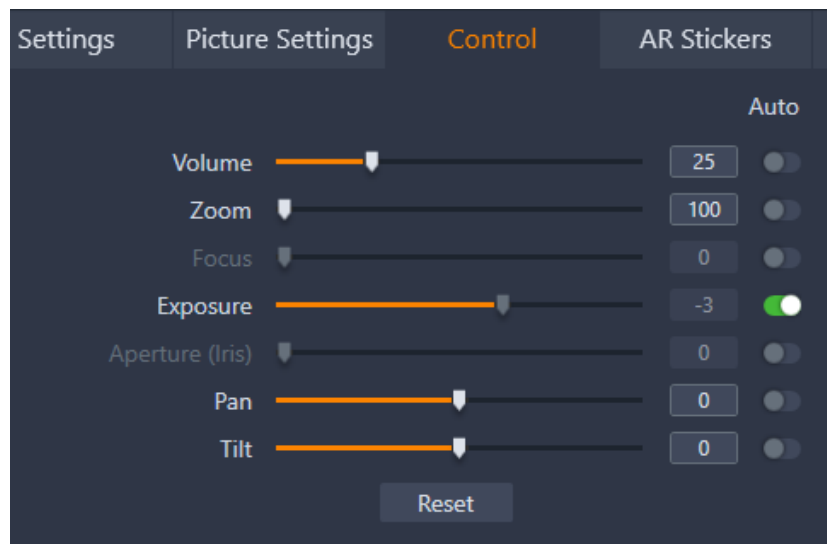
Recommended settings for the Screen recording is:





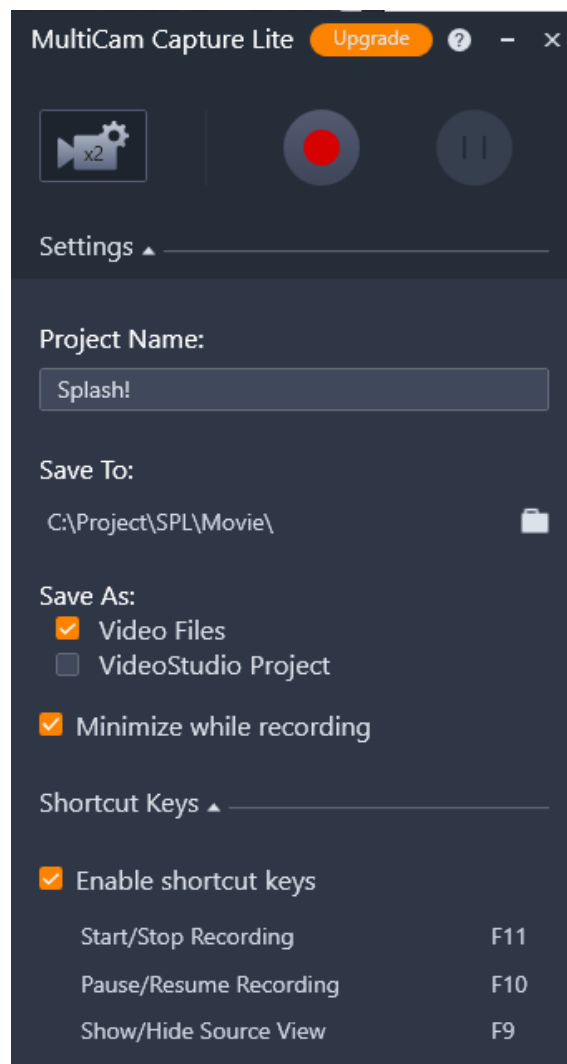
Recommended settings for the Webcam recording is:





4.3 Start and Stop recording

Use F11 for both operations. The project name will be used as sub folder for the recording.



5 Screenplayer

5.1 Basics

The Screenplayer is an intuitive basic mixing application that allows the concurrent interactive play between various sound files offline and online. It doesn't require any special education and is therefore suitable for immediate performance. However, its configuration offers rich possibilities, which can be used for highly customized applications and are described in the following manual.

The current implementation of the Screenplayer is an HTML5 prototype application running best on **Chrome** and **Firefox**.

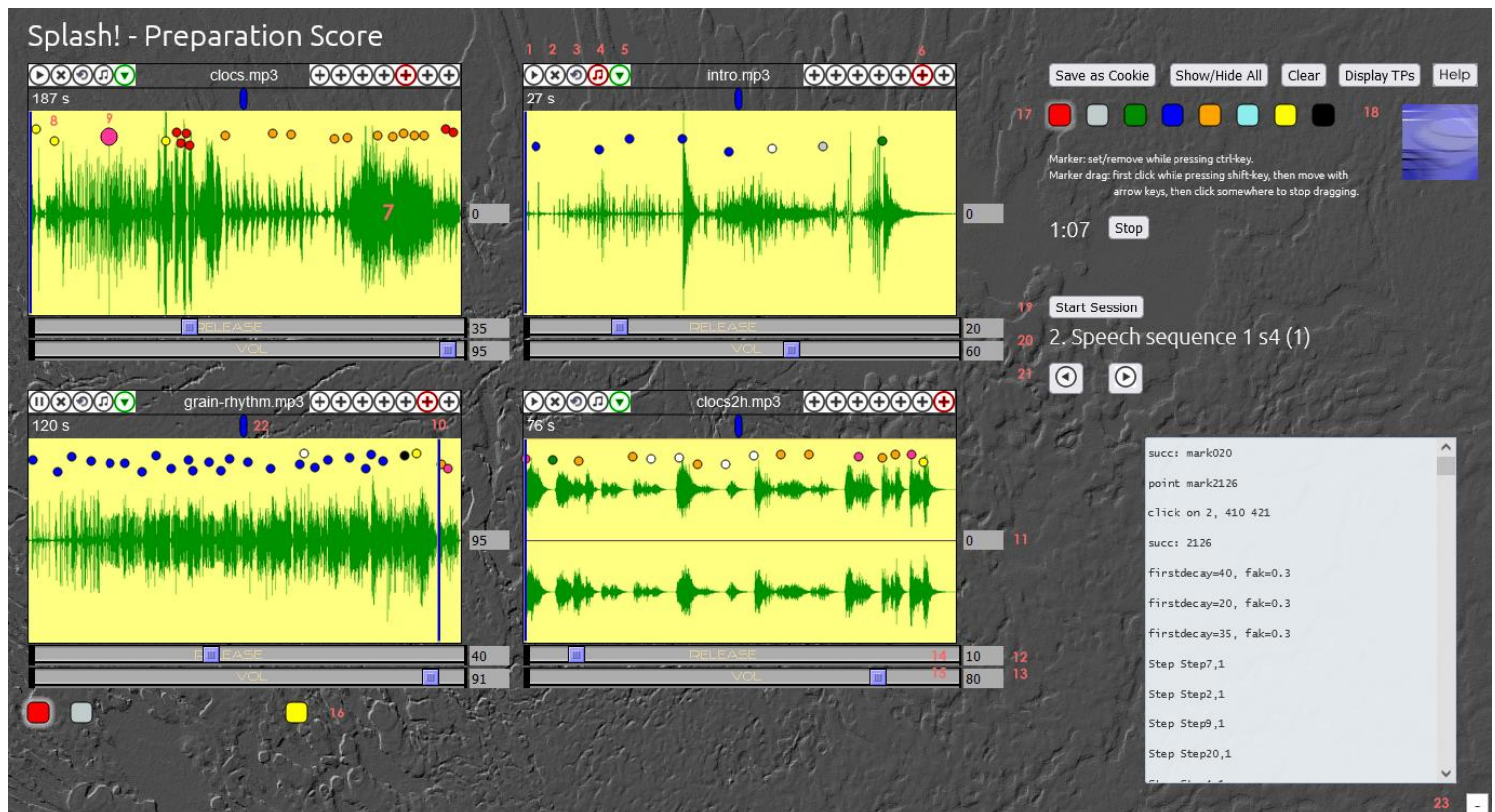
It is suitable to be used on a local computer but can be also easily operated as a web page. For the latter purpose, **mp3** sound resources should be used (however, **wav** is also possible).

The Screenplayer consists of **4 window panes** into which various sound files with their adjacent sound graphics (free of choice) can be loaded (up to 7 each).

Colored markers – so called **touchp(o)ints** – can be placed onto the sound graphics in order to memorize significant sound entry points. When clicking the sound graphics the sounds are being played together with a pre-defined envelope that also can be changed during play.

Volume level is basically controlled by use of the **vertical axis**.

5.2 Play functions



1. **Pause/Play:** hard stop, resuming play
2. **Stop:** hard stop, sets cursor (10.) back to soundfile start
3. **Loop:** continous playing from start after ending soundfile
4. **Persistenz:** click into another window will not stop soundfile play
5. **Fade out:** sound will stop with a predefined delay, the delay can be changed with the Delay slider (12.)
6. **Soundfile selection:** current sound (if there) will be interrupted and selected preloaded soundfile will be loaded into the pane
7. **Click anywhere** onto the sound graphic: Sound starts to play with predefined Fade in, sounds of other windows will end with predefined Fade out, if Persisteny (4.) is not set. Vertical axis sets the desired volume (watch slider).
8. **Touchp(o)int:** Can freely be set by using Ctrl+Click using the color chosen in the touchp(o)int Color Bar (17.). A touchp(o)int can be modified as follows:
 1. **Delete:** Ctrl+Click again onto the touchp(o)int
 2. **Move:** first click while pressing Shift-key, then move with arrow keys, then click anywhere to stop dragging
 3. **Highlighted successor:** Each touchp(o)int can have one successor in one of the 4 window panes which will be highlighted (large display, see Highlighted touchp(o)int (9.). To create a touchp(o)int successor Shift+Click the touchp(o)int and then the touchp(o)int successor, then click anywhere to stop this mode
 4. **Executing a Score step:** A touchp(o)int can jump to a Score step to execute further functions, such as:

- ◆ Loading new soundfile(s) (6.)
- ◆ Toggling Loop (3.) and/or Persistenz (4.) status
- ◆ Highlighting another touchp(o)int (9.)
- ◆ Activating another one of the 7 touchp(o)int sets (16.)
- ◆ Displaying a step text label (20.)

5. **Export:** Displaying the complete touchp(o)int set (18.) that is currently chosen in the touchp(o)int set bar (16.). The touchp(o)int set can then be copied into the sample.js configuration file (see chapter 2.3.3).



Firefox will allow a direct copy, with Chrome the adjacent cookie need to be found to be able to copy the text.

9. **Highlighted touchp(o)int:** Using a chain of highlighted touchp(o)ints allow the user to follow a predefined course of action. However, the decision about the time distance between the use of the touchp(o)ints is still free. For creation see 8.3.

10. **Sound cursor:** Displays graphically current sound position.



Clicking the sound cursor does not allow the recognition of a touchp(o)int underneath!

11. **Sound position and Fade out indicator:** Displays numerically current sound position on a scale from 0 to 100. The absolute time duration of a soundfile is displayed above the left upper corner of the window pane.

The field turns red for the duration of the predefined Fade out period of the sound.

12. **Release indicator:** Displays numerically current Fade out values on a scale from 0 to 100. The Fade out values are displayed in tens of seconds, i.e., number 50 displays a Fade out of 5 seconds. Fade in is by default set to 30% of the Fade out, i.e. 1500ms in case of Fade out = 50. This proportion can be changed using the configuration file sample.js (for syntax see chapter 2.3.2).

13. **Volume indicator:** Displays numerically current volume values on a scale from 0 to 100.

14. **Release fader:** Controls the Fade in and Fade out values in parallel according to the values set in the configuration file samples.js. Predefined values of the soundfile setting are moving the fader that can be changed manually afterwards.

15. **Volume fader:** Controls the volume of the soundfile according to the values set in the configuration file samples.js. Predefined values of the soundfile setting and – in addition – each vertical position are moving the fader that can be changed manually afterwards to continuously controlling the volume while the sound is still playing, for example, to create manually a fade out.

16. **touchp(o)int set bar:** Depending on number of existing touchp(o)int sets up to 7 knobs are displayed to allow the selection of a certain touchp(o)int set. A touchp(o)int set can apply to all existing sound files, i.e., each sound file can have up to 7 different touchp(o)ints deployments. In order to change a set, just click the desired knob.



When entering, modifying or exporting touchp(o)ints, always make sure to operate on the currently displayed touchp(o)int set.

Make sure to export the changes made to a touchp(o)int set before moving to another one. Otherwise all your changes will be lost as each call for a set will reload the set from the configuration file only.

17. **touchp(o)int color set:** Select a color before applying a new touchp(o)int (see 8.).



Further colors can be applied using the syntax of the configuration file sample.js.

18. **Display TPs:** This button allows to export a touchp(o)int set. Depending on the touchp(o)int set selected in the touchp(o)int set bar (see 16.), the current complete touchp(o)ints of all windows (also the none displayed) will be shown in a text window from which the touchp(o)int set can be copied into the configuration file sample.js.

touchp(o)ints can then further be edited using the touchp(o)int setup syntax (see chapter 2.3.3), for example to change the color, to add a touchp(o)int successor or a jump to a score step.

19. **Start session:** This launches a score that is predefined in the configuration file samples.js (see chapter 2.3.4). The window display jumps immediately to step 1 which may reload sound files and other settings.

A score can either be driven by:

- ♦ a time scheduler (see score syntax in chapter 2.3.4)
- ♦ by the score navigation buttons (21.)
- ♦ by a touchp(o)int score jump (see score syntax in chapter 2.3.4)

20. **Score step display:** In order to ease the use of the system, each score step can own a short text label with short play advice or other meaningful information such as score step number, musical characteristics and more.

21. **Score navigation buttons:** The left button launches the next step of a score, the right button the former step (if existent).



Depending on the score step content, backward changes do not necessarily reproduce the state of the former step. Score steps are based on a change protocol, not on the state of the complete configuration. In order to reproduce the correct state, a score should be launched from the beginning or from some steps before, depending on the change context.

22. **Panorama slider** (not implemented): This slider – above each window pane – is currently not supported.

23. **Debug window** (at the bottom right of the screen): Displays information meaningful for the programmer, i.e. which of the touchp(o)ints are actually clicked on. It can be used to copy information for editing the configuration file, or just be hidden by clicking the minus sign located at the bottom right.

5.3 Configuration

5.3.1 Files and folders

The standard name for the HTML5 application file is **Screenplayer.htm** and is residing at the main folder of your choice.

It requires in addition:

- ◆ **samples.js** = standard file name for the application configuration, residing in the main folder and consisting of
 - ◆ Header

Contains the title of your project as `MyTitle` as follows:

```
var MyTitle = "Splash! - Preparation Score";
var MyServer = "localhost:81";
```
 - ◆ Sound file configuration
 - ◆ Touchpoint deployment
 - ◆ Score
- ◆ **audio** folder = containing custom sound files and adjacent sound graphics
- ◆ **img** folder = containing the application graphics elements
- ◆ **script** folder = containing the application script elements
- ◆ **Ubuntu-Light.ttf** file = optional; if missing, font Arial will be used

5.3.2 Sound file configuration in samples.js

Syntax

1. `var Samples0 = new Array (`
 Introduces new file configuration for each pane, with `Samples0..Samples3` for pane 1..4.
2. File set (up to 7, last entry without ,):
 - `"<sound file>" ,`
 - `"<graphic file>" ,`
 - `"<sound configuration>" ,`
3. `<sound file> = audio/<file name>`
 File need to reside in the audio folder.
4. `<graphic file> = audio/<file name>`
 File need to reside in the audio folder.
5. `<sound configuration>`
 - ◆ `s` = Standard values for Volume (80) and Release (10)
 OR individually:
`v<0..100>d<0.100[, <seconds>]>` = individual Volume and Release values.

The Attack value – usually 30% from the Release value – can be set as factor from the d value in addition, for example, 0.5, i.e. together v50, d20, 0.5, which means 1000ms in this special case (d20 represents 2000ms).

- ◆ L = set Loop
- ◆ P = set Persistenz

Example

```
var MyTitle = "Splash! - Preparation Score";
var MyServer = "localhost:81";
var Samples0 = new Array (
    "audio/cMix-grain2.mp3",
    "audio/cubaseMix-grain2.PNG",
    "S",
    "audio/clocs2b-1.mp3",
    "audio/clocs2b-1.png",
    "v90d30"
);
var Samples1 = new Array (
    "audio/the bad wolf2.mp3",
    "audio/the bad wolf2.PNG",
    "S",
    "audio/intro.mp3",
    "audio/intro.png",
    "Pv60d20"
);
var Samples2 = new Array (
    "audio/Grandma 01.mp3",
    "audio/Grandma 01.PNG",
    "S",
    "audio/grain-rhythm.mp3",
    "audio/grain-rhythm.png",
    "v87d40"
);
var Samples3 = new Array (
    "audio/clocs2h.mp3",
    "audio/clocs2h.PNG",
    "S",
```

```

    "audio/am.mp3",
    "audio/am.png",
    "v90d20"
);

```

5.3.3 Touchp(o)int Setup in samples.js

Syntax

1. Must have (technical initialization):

```
var xFrozenCookie = "<touchp(o)int>";
```

And at least one of the seven touchp(o)int sets that are navigated by the touchp(o)int set knobs or by the score syntax (see below):

```
var FrozenCookie = "<touchp(o)int> [<touchp(o)ints>] ";
```

```
var FrozenCookie2 = "[<touchp(o)ints>] ";
```

```
var FrozenCookie3 = "[<touchp(o)ints>] ";
```

```
var FrozenCookie4 = "[<touchp(o)ints>] ";
```

```
var FrozenCookie5 = "[<touchp(o)ints>] ";
```

```
var FrozenCookie6 = "[<touchp(o)ints>] ";
```

```
var FrozenCookie7 = "[<touchp(o)ints>] ";
```

2. <touchp(o)int> consists of

```
<TP ID><TP coordinates><TP color>[mark<TP ID>]
[Step<score step><seconds>]<delimiter>
```

3. <TP ID> = <pane number 0..3><window number 0..6><TP number per soundfile 0..n>

Example: 0511 = 1st pane, 6th window (soundfile), 11th touchp(o)int

4. <TP coordinates> = x<absolute x axis value>
y<absolute y axis value>

Coordinates are based on the complete screen.

Example: x219y136

5. <TP color> = c + hexadecimal RGB values #hhhhhh

Example: c#FF0000

6. mark<TP ID>

When actual touchp(o)int is being clicked, the indicated touchp(o)int successor will be highlighted (show large circle)

Example: mark0511

7. <score step[, <seconds>]> = number from 1..n, where n is the maximum of existing score steps and optional seconds can be used as a fractal number

By default, the execution of a step change is delayed by 2 seconds. This value can be overridden by the optional second entry.



It may be a good praxis to indicate the score steps with larger scores using the text display feature of the score (see below). Executing score steps from a touchp(o)int can be helpful when organizing successive window changes for continuous uninterrupted sound play.



Score steps can be executed in any order. Since they are change-based, the resulting window state will depend on the sequential context of the score step calls.

8. <delimiter> = Each touchp(o)int entry must be followed by a - minus sign.

Touchp(o)int entries are sorted by <TP ID> in ascending order.

Example

```
0511x219y136c#FFFFFFmark3350-3226x903y450c#FFA500-
163x598y163c#FFFF00Step1-251x85y425c#FFFFFFmark3324Step3,4.-
```

5.3.4 Score in samples.js

Syntax

1. Score opening, lines and closing

```
var Score = "\
<score line 1> \
.
.
<score line n> \
"
```

2. The score line consists of:

```
-[<scheduled time>] : t<TP set> [p<pane changes>] [m<TP ID set>]
[P<persistence changes>] [L<loop changes>] [x<text>]
```

- ◆ t = set the indicated TP set
- ◆ p = changes the soundfile(s) (1..7) at the indicated pane(s) (1..4)
- ◆ m = highlight the indicated touchp(o)int or touchp(o)ints
- ◆ P = toggle the persistence button on the indicated pane(s) (1..4) based on sound configuration
- ◆ L = toggle the loop button on the indicated pane(s) (1..4) based on sound configuration
- ◆ x = followed by text label

3. <scheduled time> in seconds as a fractal number

Scores should be used either with or without time schedule as a whole.

When using scheduled steps, the text label of the step is colored differently for the last 3 seconds before the change to the next step to indicate a countdown.

4. <TP set> = touchp(o)int set 1..7

Example: t2



This parameter always should be set even the identical set had already been loaded in the former step. Otherwise, highlighting of certain touchp(o)ints may not disappear when clicking the touchp(o)int.

5. <pane changes> = <pane change>[, <pane change>]*0..3

<pane change> = <pane number 1..4><window number 1..7>

Example: p10,21

6. <TP ID set> = <TP ID>[, <TP ID>]*0..n

Example: m101,2111,319

7. <persistence changes> = <persistence toggle>[, <persistence toggle>]*0..3

<persistence toggle> = 1..4

Example: P1,2

8. <loop changes> = <loop toggle>[, <loop toggle>]*0..3

<loop toggle> = 1..4

Example: L1,3,4

9. <text> = Free text label with approx. maximum of 34 characters.



Forbidden characters are minus symbol - and slash /.

Example 1 (without time schedule)

```
var Score = "\
-:t1p11,21,31,41m1010x1. Morph Sequences s1,s2\
-:t1p13,22,32m2126x2. Speech sequence 1 s4 (1)\
-:t1p43m2119x3. Speech sequence 1 s4 (2)\
-:t2p13,32,45P1,2m0214x4. Noise 1 s6 (1)\
-:t2p24P2m133x5. Noise 1 s6 (2)\
"
```

Example 2 (with time schedule)

```
var Score = "\
-2:t1p11,21,31,41m1010x1. Morph Sequences s1,s2\
-3.5:t1p13,22,32m2126x2. Speech sequence 1 s4 (1)\
-4.5:t1p43m2119x3. Speech sequence 1 s4 (2)\
-3:t2p13,32,45P1,2m0214x4. Noise 1 s6 (1)\
-2.8:t2p24P2m133x5. Noise 1 s6 (2)\
"
```

"

5.3.5 Screenplayer Customization in Screenplayer.htm

The **Screenplayer.htm** file can be customized as follows.

1. Set the topic:

Search for `<title>` and add your own project title.

Example: `<title>Splash! - Preparation Score</title>`



This title will appear in the browser title bar, NOT at the Screenplayer. To change the project name there, you must enter it at the beginning of the configuration file **samples.js**, see chapter 2.3.1.

2. Set the title color:

Search for `TitleColor`.

Example: `var TitleColor = "#789DFA";`

3. Apply logo:

Search for `logo`.

Example: ``

Place your logo in the **/img** folder and change the file name in the above line accordingly.

4. Apply background image:

Search for `background:transparent`.

Example: `background:transparent url(img/WP3.png) no-repeat 100% 100%;`

Place your background image in the **/img** folder and change the file name in the above line accordingly.

6 Playing Examples

6.1 Introduction

This project is laid out for **four** players and is being presented to the public within the framework of the festival BLAUES RAUSCHEN in September 2022 in Germany. For more information visit

<https://hartmannfriedhelm.wixsite.com/my-site/post/blaues-rauschen>.

A demonstration can be viewed at YouTube at <https://youtu.be/padoG7ohCKQ> which includes also the **copyright** information.

The project includes various playing types as described below and can be accessed and performed online as follows:

- ◆ "Blue Noise Splash": Demo-version
<https://www.cm-gallery.com/Projects/HSML/Screenplayer-b.html>.
- ◆ "Splash Noise": Festival-version used by four international artists
<https://www.cm-gallery.com/Projects/HSML/Screenplayer-n.html>.
- ◆ "Splash Noise Blues": Festival-version with four international artists and audience interaction
<https://www.cm-gallery.com/Projects/HSML/Screenplayer-s.html>.
- ◆ "Path with Glades": Version for two players
<https://www.hjflorian.de/BlueNoise22/Screenplayer.html>

The playing rules are always **specific** to the playing version. Each Screenplayer version can have its own set of rules.

6.2 Playing Rules Example 1: „Blue Noise Splash“

1. There are **four basic colors** of touchp(o)ints available: red, green, blue and yellow. Each player/party obtains one of the colors. No player should ever use the color of another player.
2. **Color shades:**
 - ◆ Normal: Clicking the touchp(o)int leads to the next set of one or more highlighted touchp(o)ints which may also include a change of sounds loaded to the panes.
 - ◆ Dark: Leads to the former score step.
 - ◆ Light: Touchp(o)int set remains while further touchp(o)ints may be clicked/touched that may or may not disappear in which case it can be clicked again.
 - ◆ Colored: Jumps a few steps in a score.
3. As a general rule, **no player should click twice** but letting another player click the next touchp(o)int first – unlike there is no other option.
4. The **choice of a touchp(o)int** within a set of common colors **is** completely **free** and solely in the musical responsibility of the color's player. The same is valid for the **response time**. It may be a good advice to change the response time from time to time, ranging from immediate reply down to several seconds of response, based on the musical context.

5. This project is score-based (see chapter 5.3.4.). Therefore, **only touchp(o)ints** need to and should be operated. No other Screenplayer function should be used during the performance.

Average duration of this version: 6-8 minutes.

6.3 Playing Rules Example 2: “Splash Noise”

This play version requires more experienced artists as it doesn't make any use of touchp(o)ints.

1. Each sound agent gets **one** of the four windows as his own playing window area and uses the sound panes for individual free sound choices based on playing experience with the sounds.

This way, viewers are still able to identify who is currently contributing to the piece.
2. Beside the click/touch actions on the screen, each player decides freely when to
 - ◆ **change** on of the 7 **sound panes** to invoke different sounds,
 - ◆ involve the **Persistence** function to cause continuous play,
 - ◆ execute the **Fade out** action for his sounds currently playing.

Since one cursor is shared by four players without further coordination, only momentarily actions should be performed as the use of sliders would often be interrupted by other players actions.

3. The playing team agrees on a **timeframe** of the project and finishes the piece by watching the project clock accordingly.

6.4 Playing Rules Example 3: “Splash Noise Blues”

This version allows the audience to be involved.

1. Each sound includes 9 touchp(o)ints at different vertical positions which indicate various **loudness** positions.
2. There are **four colors** deployed to guide the audience with regard to the sound characteristics that can be expected, supported by the sound graphics as well:
 - ◆ Red: Speech like or otherwise short musical phrases that can be used for kind of melodic expressions
 - ◆ Green: Long lasting sounds like drones
 - ◆ Blue: Rhythmical patterns
 - ◆ Yellow: Special effects
3. The audience gets **laser pens** to point to the sound screen displayed at the location of the live performance where also the other players over the Internet are displayed.

The sound agent who is present at the live performing site executes any time a new touchp(o)ints on the performance computer as soon as at least three laser dots from the public are pointing to the same touchp(o)int. Every touchp(o)int is used only for one time and will remain in small display.



Video dedection software could be used optionally to interpret the audience actions automatically.

4. The other three players identify the touchp(o)int choices by the audience by means of the change **from large to small** display with every click.
5. The onsite and the remote sound agents changing **sound panes**, performing **sound persistence** and **fade out** actions along with the touchp(o)int-choices by the audience.
6. The piece finishes while **removing** all still available touchp(o)ints either automatically after 4 minutes or by executing the gray colored touchp(o)int group button at the bottom at the second position.

6.5 Playing Rules Example 4: "Path with glades"

This version is best executed with two players. They follow predetermined paths through a sound landscape, but also encounter glades that allow them to roam freely and to linger at their own pace.

1. **Sound activation:** The players alternately activate the touchp(o)ints.
2. **Paths:** The paths consist of red and green touchp(o)ints. Activating a touchp(o)int highlights its successor. One player serves the red dots, the other player the green dots. Bright touchp(o)ints determine the beginning of each path.
3. **Glades:** The colors of the glades consist of gray and yellow touchp(o)ints. All points of a glade are highlighted. The players choose the individual touchp(o)ints as they wish. Dark touchp(o)ints mark the transition to a new glade or path. The choice to activate such a touchp(o)int can be made by any of the players at any time.
4. **Exceptions:** In step 8 of the score, the 4th touchp(o)int from the right in Pane 2 (top right) only allows one successor.

In step 11, the 2nd and 3rd touchp(o)int of the 3rd pane only allow two touchp(o)ints in pane 2, clicking on it again allows the 2nd and 3rd as well as the 1st touchp(o)int in pane 3. The 1st touchp(o)int opens again all gray touchp(o)ints.

Credits

The second half of the sound "past.mp3" consists of excerpts from "Just another Odyssey" (2003) and "The bad wolfe3" (2014) by Friedhelm Hartmann.

The setup and all the other sounds are from H.-J. Florian.

(C) Copyright 2022 by Heinz-Josef Florian