# Dugout

|  |  |  |
| --- | --- | --- |
| Autor | Datum | Změny |
| Daniel Štreit | 22.4.2016 | Převod do .doc, přidány obrázky |
| Radim Buchtela | 24.4.2016 | Upraveny jména scén, návrh komunikačního protokolu |

Požadavky na grafiku.

* Název hry: Dugout
* Jazyk: Angličtina
* Rozlišení pro TV: 1280x720px
* Původní zdroj: [\\MSTORAGE002\Dugout\](file:///\\MSTORAGE002\Dugout\)

Aplikační ikona

* Aplikační ikona v rozlišení 192x192px
* Viz. zde <https://sdk.metrological.com/design-guide#iconsize>

# TV screen

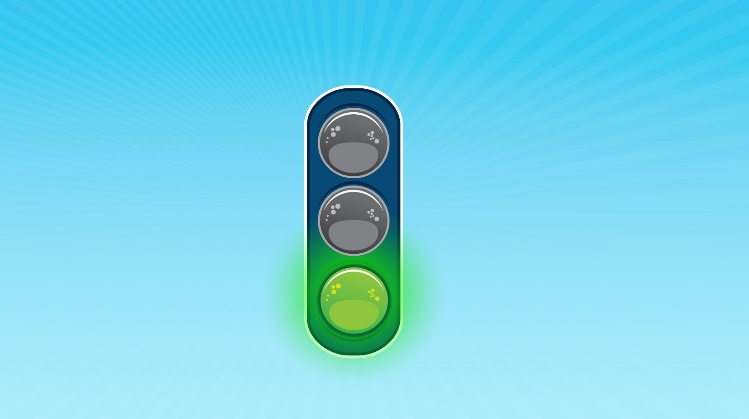
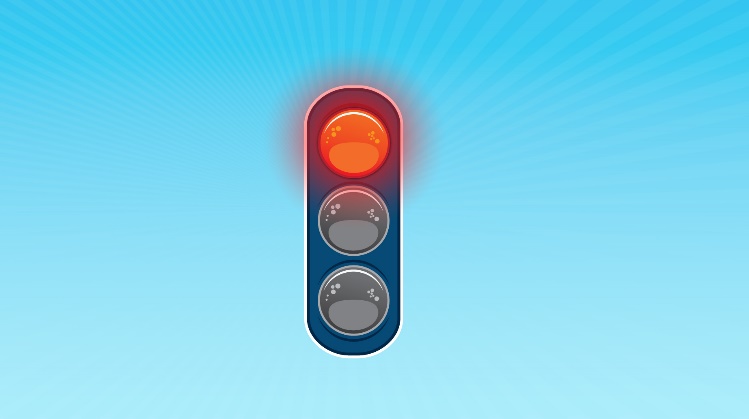
## Splashscreen [scene]

* Rozlišení: 1280x720px
* Zdoj: ./mokcups1/Dugout\_\_02.jpg
* Vytvořit:
  1. Pozadí
  2. Logo

## Home [scene]

* Rozlišení: 1280x720px
* Zdroj: ./mokcups1/Dugout\_\_05.jpg
* Bude potřeba umístit QR code – nalevo od seznamu hráčů (šířka seznamu hráčů bude oproti podkladu menší), pokud jsou přihlášeni všichni čtyři hráči, QR code může být schován a volné místo se může využít pro zvětšení seznamu hráčů.
* V tabulce bude pouze barevná žížala a jméno
* Vytvořit:
  1. Background (bez loga, bez tabulky hráčů, bez žlutých ikon)
  2. Logo
  3. Čtyři ikony žížal
  4. Ikona otazník (about) + focusovaný stav
  5. Ikona kolečko (setting) + focusovaný stav
  6. Navrhnout ikona help + focusovaný stav
  7. Navrhnout tlačítko start + udělat focusovaný stav (spíš bude ovládáno přes mobily)

## Countdown [scene]

* Rozlišení: 1280x720px
* Zdroj: ./mokcups1/Dugout\_\_SEMAFOR\_\_01.jpg,./mokcups1/Dugout\_\_SEMAFOR\_\_02.jpg,./mokcups1/Dugout\_\_SEMAFOR\_\_03.jpg
* Vytvořit:
  1. Pozadí
  2. Semafor - červená
  3. Semafor - žlutá
  4. Semafor - zelená

## Playground [scene]

* Rozlišení: 1280x720px
* Zdroj: ./mokcups1/Dugout\_\_logo\_\_08\_\_podzemi.jpg
* Vytvořit:
* Čtyři ikony žížal (použité nahoře u skóre)
* Čtyři ikony žížal ve hře
* Obrázek hrobeček
* Bonusy (kladné/záporné – zelené/červené):
  1. SpeedUp
  2. SpeedDown
  3. BetterTurning
  4. Lékárnička – možnost přejít jednu stopu
  5. Mince – navýšení skóre

## Podium [scene]

* Rozlišení: 1280x720px
* Zdroj: ./mokcups1/Dugout\_\_vyherniprostredi\_\_02.jpg
* Vytvořit:
  1. Pozadí (bez položek nadpis, jména, score a jejich podkladu, bez žížal)
  2. Čtyři žížaly
  3. Pokud bude hráčů méně – bude pozice prázdná

## Settings [scene]

* Rozlišení: 1280x720px
* Navrhnout settings cenu (vycházet by měla z.: /mokcups1/navrh\_roman.png, ale na celou obrazovku)



* obrazovka by měla obsahovat:
  1. Difficulty – easy/normal – rychlost pohybu žížaly
  2. Music – on/off
  3. SoundEffects – on/off
  4. SelectMapColor – možná více pozadí při hře – pouze jako vylepšení

## About [scene]

* Umístit QRcode (odkaz na Mautilus) + logo Mautilus

## Info [scene]

* Popisek ovládání hry

Možná spojit About scene s Info scene.

# Second Screen na mobilu

## Navázání spojení

* Loading message

## Přihlášení – Uživatel musí zadat jméno a potvrdit

* Input pro jméno
* Potvrzovací tlačítko

## Uživatel přihlášen

* Na mobilu se zobrazí tlačítko Start
* Tlačítko Start

## Zahájená hra

Uživatel ovládá směr žížaly dvěma tlačítky left/right (obrazovka mobilu bude rozdělena na dvě poloviny - ovládání směru žížaly.)

* Ikona left a ikonar ight

## Stav smrti

* Ikona hrobečku (zbarveného kříže)

## Zobrazení skóre na konci hry (když všichni dohráli)

* Zobrazit skóre se stupni vítězů

## Otázka pokračovat v nové hře

* Tlačítka ano – ne

# Komunikace mezi mobilními klienty a TV app

## Use case – Přihlášení uživatele

barvicky = [cervena, zelena, zluta, modra]

1. Zobrazení Home scene

* Zobrazen QR kód
* Defaultně bude zobrazen seznam 4 hračů – beze jmen (to znamená, že žádný hráč není připojen)

1. Přihlášení hráče
   * Uživatel naskenuje QR kód přes svůj mobil
   * Na svém mobilu načte webovou aplikaci
   * Webová aplikace se připojí k serveru v místní síti
   * Server si uloží uživatele a jeho hash, přiřadí mu indexPlayer = {1,2,3,4}, barvicku = barvicky[indexPlayer - 1], defaultně jméno = “Player\_index” a stav=’pripojen’ a provede:
     1. Informuje TV app o přihlášení uživatele zasláním zprávy (‘pripojen‘, indexPlayer/barvicka, jmeno). TV vykreslí uživatele do seznamu hračů. Ve stavu připojen.
     2. Informuje Mobil app o úspěšném připojení zasláním zprávy (‘pripojen‘, indexPlayer/barvicka, jmeno).
   * Mobilní app přijme zprávu ‘pripojen‘ a vykreslí ‘prihlaseni‘ scene, kde uživatel vyplní jméno a potvrdí – přejde do stavu “pripraven”. A informuje o tom server zasláním zprávy (‘pripraven’, jmeno).
   * Server přijme zprávu (‘pripraven’, jmeno), uloží si stav a jméno a provede:
     1. Informuje o tom TV app zasláním zprávy (‘pripraven’, indexPlayer, jmeno), TV app u daného hráče zobrazí stav ‘pripraven‘ a aktualizuje jméno.
     2. Zkontroluje stav všech přihlášených hráčů, pokud jsou všichni ve stavu ‘pripraven‘ informuje o tom všechny hráče zasláním zprávy (‘all\_ready‘).

* Mobil app přijme zprávu ‘all\_ready‘ a zobrazí tlačítko ‘start‘
* Mobil app uživatel klikne ‘start‘ a informuje o tom server zasláním zprávy (‘start\_game‘).
* Server přijme zprávu ‘start\_game‘ a provede:
  + 1. informuje o tom TV app zasláním zprávy ‘start‘, TV app zobrazí semafor (odpočet).
    2. Informuje o tom všechny přihlášené hráče zasláním zprávy ‘show\_controls‘.
  + Mobil app přijme zprávu ‘show\_controls‘, zobrazí šipky pro ovládání hry a zavibruje.
  + Mobil app, uživatel ovládá pohyb žížaly a informuje o tom server zasláním zprávy (‘left‘) nebo (‘right‘),
  + Server přijme zprávu (‘left/right‘) a informuje o tom TV app zasláním zprávy (‘left/right‘, indexPlayer)

1. Konec hry

* TV app zobrazí Konečné pořadí a přejde zpět na úvodní scénu. Přihlášení hráči zůstávají přihlášeni.
* TV app informuje server zasláním zprávy (‘new\_game‘) a server přijme zprávu (‘new\_game‘) u všech hráčů si nastaví stav ‘prihlasen‘ a pošle zprávu všem hráčům zprávu (‘continue‘).
* Mobil app přijme (‘continue‘) a zobrazí ‘chcete pokračovat? ‘ s tlačítky ‘ano‘/‘ne‘.
* Mobil app, uživatel klikne ‘ano‘ a informuje o tom server zasláním zprávy (‘ready‘).

A pokračuje viz. výše

* Mobile app, uživatel klikne ‘ne‘ a informuje o tom server zasláním zprávy (‘exit‘). Server to zpracuje, smaže si hráče ze seznamu a informuje o tom TV app, ta vykreslí prázdného hráče ze seznamu.

## Komunikační protokol

* Komunikace bude realizována přes prostředí [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room), které poskytuje Metrological framework
* [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room) - jedná se o broadcastový typ komunikace, kdy jednotliví klienti připojeni do ‘room‘ přijímají všechnu komunikaci (klient odešle zprávu a ta je doručena všem ostatním klientům připojených do [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room)). Všichni klienti jsou si rovni. Klient nemůže vykopnout jiného klienta z [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room).
* Proto, abychom byli schopni zdárně realizovat komunikaci přes [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room), musíme si nadefinovat role klientů a navrhnout vnitřní komunikační protocol, kterým budou komunikovat klienti uvnitř [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room).

### Role klientů

* + mobil\_klient (0 až n) – představuje uživatele připojeného přes mobilní app.
  + tv\_supervisor (0 nebo 1) – představuje TV app, která organizuje hru a komunikuje s mobil\_klienty.

### Protokol

* Jelikož [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room) představuje broadcastový typ komunikace, musíme zde také zavést mechanismus, jak adresovat zprávu konkrétnímu klientu.
* Základní zpráva je v nasledujícím formátu:

{event: ‘nazev\_udalosti‘, clients: [], attrs: {}}

* + *event* – název události.
  + *clients* – pole obsahující id klientů, kterým je zpráva adresována. Pokud je pole prázdné, je zpráva adresována všem.
  + *attrs* – doplňkové atributy.

Příklad:

{event: ‘tv\_accepted‘, clients: [‘da9xe‘], attrs: {color: ‘red‘, name: ‘Player1‘}}

nebo

{event: ‘tv\_rejected‘, clients: [‘da9xe‘], attrs: {error: { code: ‘300‘, message: ‘přihlášeni 4 hráči‘}}

Pozn. Zprávy událostí jsou kódovány jako JSON objekty.

## Seznam událostí

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| událost | clients | attributy | význam | reakce |
| cl\_join | [] |  | Klient se chce připojit ke hře | TV Supervizor odpoví událostí tv\_join\_accepted nebo tv\_join\_rejected |
| tv\_accepted | [id\_klienta] | color {String}  name {String} | TV potvrdila přijetí hráče do hry a pošle mu barvu a přiřazené defaultní jméno ve tvaru PlayerX | Klient zobrazí přijaté default jméno a barvičku a čeká než uživatel potvrdí, že je ready hrát (může i změnit jméno) |
| tv\_rejected | [id\_klient] | error {Object}  error.code {Number}  error.message {String} | TV zamítla přijetí hráče, v parametru error bude informace proč je zamítnut |  |
| cl\_ready | [] | name {String} | Klient pošle, že je připraven na zahájení hry. Posílá jméno, co se má zobrazit u hráče na TV | TV Supervizor nastaví jméno hráče. A překreslí jméno a stav na ready u daného hráče. Čeká se dokud všichni nejsou ready. |
| tv\_all\_ready | [ids všech co jsou ready] | num\_players {Number} | TV posílá, když jsou všichni přihlášení hráči ve stavu ready. V parametru num\_players posílá počet přihlášených hráčů, tj 0 < n <= 3. Pokud je počet 4, tak se tato událost neposílá a rovnou se posílá událost „tv\_show\_controls“. | Klient zobrazí tlačítko start. Kterým mohou zahájit hru. |
| tv\_not\_all\_ready | [ids všech co jsou ready] |  | TV posílá, pokud se přihlásil nový hráč po odeslání „tv\_all\_ready“ | Klient zobrazí zprávu čeká se ještě na hráče |
| cl\_start\_game | [] |  | Klient pošle příkaz pro zahájení hry. Tuto událost může poslat jen pokud mu byla doručena předchozí událost „tv\_all\_ready“ | Klient čeká, dokud mu nedojde událost „tv\_show\_controls“ |
| tv\_show\_controls | [ids všech hrajicich] |  | TV posílá při startu hry | Klient zobrazí tlačítka pro ovládání pohybu žížaly |
| cl\_turn\_left | [] |  | Klient posílá pro ovládání pohybu žížaly – zatočení vlevo | TV Supervizor předá informaci hernímu enginu |
| cl\_turn\_right | [] |  | Klient posílá pro ovládání pohybu žížaly – zatočení vpravo | TV Supervizor předá informaci hernímu enginu |
| tv\_death | [id\_klient] |  | TV posílá při smrti hráče | Klient zobrazí hrobeček |
| ~~tv\_score~~ | ~~[ids vsech hrajicich]~~ | ~~score {Array}~~  ~~score[client] {Object}~~  ~~client.id {String}~~  ~~client.score {Number}~~  ~~client.rank {Number}~~ | ~~TV posílá výslednou listinu při ukončení hry~~ | ~~Klient zobrazí své skóre~~ |
| tv\_continue | [ids všech hrajících] |  | TV posílá po skončení hry, hráči zůstávají přihlášeni. A čeká pokud chtějí pokračovat ve hře tím, že pošlou událost „ready“ | Klient zobrazí zprávu ‘chete pokračovat v nové hře? ‘ a zvolí buď ‘ano‘ a odešle událost ‘cl\_ready‘. Nebo zvolí ‘ne‘ a odešle událost ‘cl\_exit‘ |
| cl\_exit | [] |  | Klient posílá, pro odhlášení ze hry. | TV Supervizor si ho smaže ze seznamu. A v závislosti na počtu hráčů a jejich stavu může odeslat události např. „tv\_all\_ready“ nebo vyčká dokud nejsou všichni ready |
| dbg\_message | [] |  | Posílání zpráv mezi klienty a tv pro debugovací účely. | Přijata zpráva musí být vypsána do konzole. |

Pokud se klient **odpojí** z [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room), je to totožné situaci, jako by odeslal událost ‘**cl\_exit**‘.

Pokud se klient **připojí** do [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room), je to totožné situaci, jako by klient odeslal událost ‘cl\_join‘. Tedy pokud TV supervisor detekuje připojení nového klienta do [MAF.Room](https://sdk.metrological.com/sdk/MAF.Room), pak reaguje stejně jako by klient poslal událost ‘**cl\_join**‘.