

Documentação GLBoard

GLBoard é um conjunto de ferramentas que lhe auxiliam no processo de captura e análise dados de jogos educacionais.

Sobre GLBoard

Muitos jogos, em especial os educacionais não possuem um sistema de captura e análise de dados, acreditamos que isso ocorra pela complexidade de se implementar tal sistema. A falta que esse processo de captura/análise faz é grande, pois sem isso é difícil tentar comprovar que o jogo educacional realmente está sendo eficiente no seu objetivo de auxiliar na aprendizagem de determinado assunto.

Além disso outra dificuldade que foi observado através de pesquisas, é que muitos desenvolvedores de jogos educacionais não sabem ao certo quais dados devem capturar.

Pensando nisso foi desenvolvido um conjunto de ferramentas, para auxiliar nesse processo.

Ao todo existem quatro ferramentas que serão melhores descritas abaixo.

- GLBoard-Template
- [GLBoard-API](#)
- [GLBoard-Package-Unity](#)
- [GLBoard-WEB](#)

Caso deseje entender na prática como integrar o GLBoard com o seu jogo basta assistir essa [playlist](#) de quatro vídeos.

GLBoard Modelo de Captura de Dados - V 1.0.0

Antes de tudo é necessário entender o modelo que foi proposto para captura de dados, esse modelo é a base de todas as ferramentas que compõe o GLBoard.

Esse modelo possui uma estrutura [JSON](#), que busca capturar os dados dos jogadores, tanto dados de perfil quanto dados de seu percurso no jogo. Logo abaixo é possível visualizar um modelo genérico preenchido com dados de um determinado jogador. Caso não consiga compreender o JSON por completo, não se preocupe pois cada um dos campos do JSON vai ser detalhado em seguida.

```
{  
  "player_data": {
```



```
"day_birthday": "21/04/1999 00:00:00",
"gender": "FEMININO",
"id": "Ana",
"name": "Ana"
},
"game_data": {
  "custom_report": "Esse usuario geralmente tem problemas em fases longas...",
  "number_phases": 4,
  "date_last_login": "21/04/1999 00:00:00",
  "phases_unlocked": 1,
  "player_minutes_in_game": 1200,
  "phases": [
    {
      "phase_id": "Fase 0",
      "status": "FINALIZADA",
      "sections": [
        {
          "conclusion": "DERROTA",
          "dateTimeFinish": "03/10/2021 22:03:05",
          "dateTimeStart": "03/10/2021 21:43:05",
          "finalized_challenges": [
            "Derrotou Inimigo 1",
            "Encontrar tesouro"
          ],
          "path_player": [
            "Plataforma 1",
            "Plataforma 2",
            "Plataforma 3"
          ],
          "performance": 30,
          "route_image_b64": "BASE 64"
        },
        {
          "conclusion": "DERROTA",
          "dateTimeFinish": "03/10/2021 22:03:05",
          "dateTimeStart": "03/10/2021 21:43:05",
          "finalized_challenges": [
            "Derrotou Inimigo 2",
            "Encontrar tesouro",
            "Ver Mapa"
          ],
          "path_player": [
            "Plataforma 1",
            "Plataforma 2",
            "Plataforma 3"
          ]
        }
      ]
    }
  ]
}
```

```
    ],
    "performance": 30,
    "route_image_b64": "BASE 64"
  },
  {
    "conclusion": "VITORIA",
    "dateTimeFinish": "03/10/2021 22:03:05",
    "dateTimeStart": "03/10/2021 21:43:05",
    "path_player": [
      "Plataforma 1",
      "Plataforma 2",
      "Plataforma 3"
    ],
    "performance": 30,
    "route_image_b64": "BASE 64 IMAGE"
  }
]
},
{
  "phase_id": "Fase 1",
  "status": "NAO_FINALIZADA"
},
{
  "phase_id": "Fase 2",
  "status": "NAO_FINALIZADA"
},
{
  "phase_id": "Fase 3",
  "status": "NAO_FINALIZADA",
  "sections": [
    {
      "conclusion": "DESISTENCIA",
      "dateTimeFinish": "03/10/2021 22:03:05",
      "dateTimeStart": "03/10/2021 21:43:05",
      "path_player": [
        "Plataforma 1",
        "Plataforma 2",
        "Plataforma 3"
      ],
      "performance": 30,
      "route_image_b64": "BASE 64 IMAGE"
    }
  ]
},
},
{
```

```
    "phase_id": "Fase 4",
    "status": "NAO_FINALIZADA"
  }
],
}
```

PLAYER DATA

```
"player_data": {
  "day_birthday": "21/04/1999 00:00:00",
  "gender": "FEMININO",
  "id": "Ana",
  "name": "Ana"
},
```

Os dados de **Player Data** estão relacionados diretamente ao perfil do jogador.

Campo	Tipo	Descrição
day_birthday	STRING(FORMAT DATETIME)	Data de nascimento do jogador.
gender	STRING(FEMININO, MASCULINO, OUTROS)	Gênero do jogador, ao invés de "OUTROS" pode especificar o gênero.
id	STRING	ID único do jogador, no seu jogo não deve haver outro jogador com o mesmo ID.
name	STRING	Nome do jogador, pode existir outro jogador com o mesmo nome.

GAME DATA

```
"game_data": {
  "custom_report": "Esse usuario geralmente tem problemas em fases longas...",
  "number_phases": 4,
  "date_last_login": "21/04/1999 00:00:00",
  "phases_unlocked": 1,
  "player_minutes_in_game": 1200,
  "phases": []
}
```

São muitos os dados que compõe o **Game Data**, mas de forma geral esses dados estão relacionados aos dados gerados pelos jogadores durante sua *gameplay*.

Dentre os dados gerais do *game_data*, constam:

Campo	Tipo	Descrição
custom_report	STRING	Neste campo fica a critério do desenvolvedor dar um feedback mais específico sobre o jogador dentro do jogo por exemplo: "Esse jogador tem dificuldades com fases que possuem muitas informações". Não serão realizadas análises sobre este <i>custom_report</i> porém esse campo é importante para informa detalhes do jogador que o modelo não abrange.
number_phases	INT	Quantidade de fases que o jogo possui.
phases_unlocked	INT	Quantidade de fases que o jogador finalizou até o momento.
player_minutes_in_game	INT	Minutos que o jogador possui de jogo. Contabiliza desde o minuto que o jogador entrou até o minuto que saiu.
phases	ARRAY(PHASES)	Possue um array de phase, relacionado a cada fase do jogo, cada fase possui suas informações.
date_last_login	STRING(DATE TIME)	Neste campo deve ser informado o horário ultimo login do jogador

O **Game Data**, possui uma estrutura que é bastante comum em jogos:



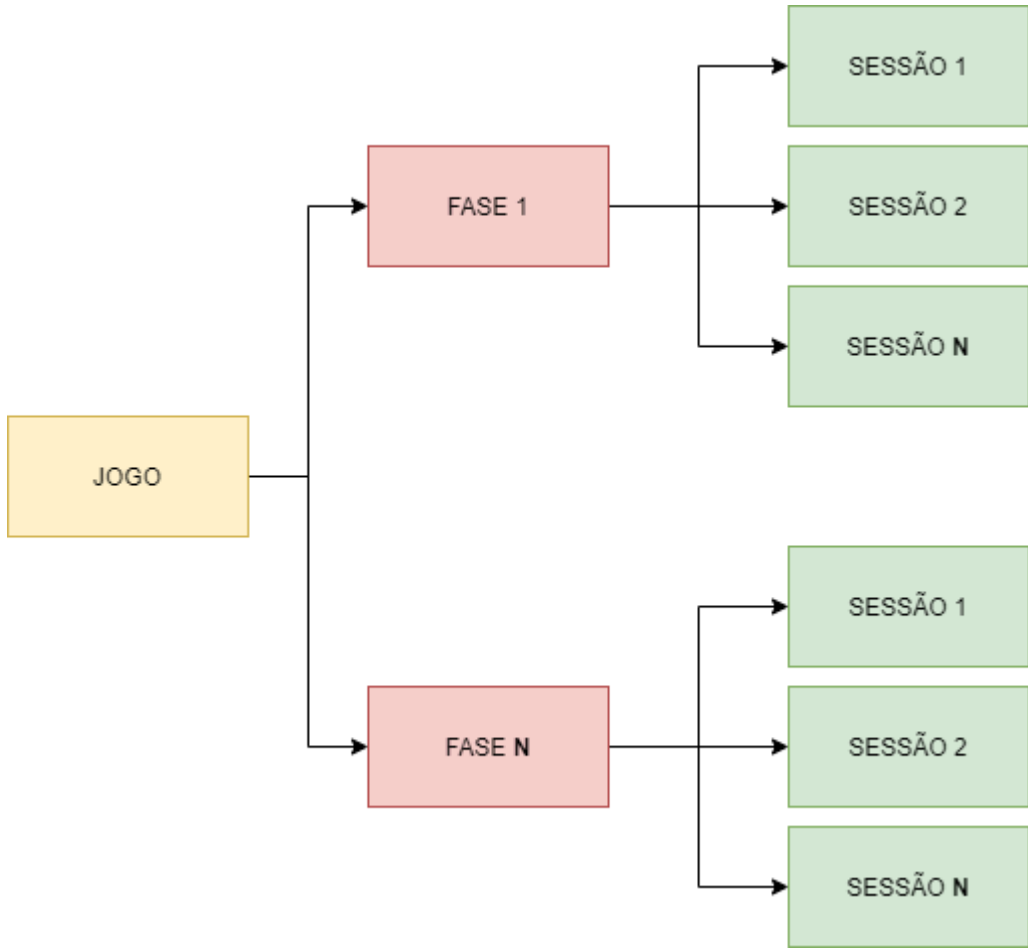


DIAGRAMA DE FASES

No diagrama acima é possível visualizar que um jogo pode ser constituído por 1:N Fases e cada fase pode conter 1:N Sessões. Não necessariamente o jogo precisa ter essa divisão para se encaixar no modelo genérico, por exemplo um jogo contínuo que não tem interrupções também possui fases que não são explicitas, mas nesses jogos o percurso de um *checkpoint* para outro pode ser interpretado como uma fase e as tentativas realizadas entre esses *checkpoints* pode ser compreendido como sessões.

PHASE DATA

```
{
  "phases": [
    {
      "phase_id": "Fase 0",
      "status": "NAO_FINALIZADA",
      "sections": [],
    },
    {
      "phase_id": "Fase 1",
      "status": "NAO_FINALIZADA",
      "sections": [],
    }
  ]
}
```

Um objeto do tipo Phase é o que constitui um array de Phases no campo *phases* do JSON. Dentre os dados gerais de uma Phase tem:

Campo	Tipo	Descrição
phase_id	STRING	ID unico de uma fase, não pode existir outra fase com este nome, não necessariamente uma fase precisa ser intitulada como "Fase 1"/"Fase 2"/"Fase 3"... O phase_id pode ser realmente o nome da fase, por exemplo "Fase dos caminhos perdidos", "Fase da viagem"...
status	STRING(NAO_FINALIZADA, FINALIZADA)	O status de uma fase pode ser finalizada ou não finalizada, caso o jogador chegue ao final com êxito a fase deve ser marcada com finalizada.
sections	ARRAY(SECTION)	Possue um array de <i>sections</i> , relacionado a cada fase do jogo, cada sessão possui suas informações.

SECTION DATA

```
{
  "conclusion": "DERROTA",
  "dateTimeFinish": "03/10/2021 22:03:05",
  "dateTimeStart": "03/10/2021 21:43:05",
  "finalized_challenges": [
    "Derrotou Inimigo 2",
    "Encontrar tesouro",
    "Resolveu equação",
    "Ver Mapa"
  ],
  "path_player": [
    "Plataforma 1",
    "Plataforma 2",
    "Plataforma 3"
  ],
  "performance": 30,
  "route_image_b64": "BASE 64"
}
```

Um objeto do tipo Section é o que constitui um array de *Sections* no campo *sections* de uma determinada Phase.

DEVE SER COMPREENDIDO COMO SESSÃO UMA TENTATIVA REALIZADA EM UMA FASE

Dentre os dados gerais de uma Section tem:

Campo	Tipo	Descrição
conclusion	STRING(DERROTA, VITORIA, DESISTENCIA)	Conclusão de uma sessão, essa conclusão pode ser VITORIA, DERROTA ou DESISTENCIA. Este é o resultado da tentativa do jogador.
dateTimeStart	DATETIME	Tempo de inicio da sessão.
dateTimeFinish	DATETIME	Tempo de termino da sessão.
finalized_challenges	ARRAY(STRING)	<p>Array contendo os desafios que o jogador finalizou durante a sessão. Neste campo é interessante informar os desafios que o jogador venceu ou até <i>checkpoints</i> que alcançou.</p> <p>Quanto mais detalhado for esse campo melhores são as análises posteriores que podem ser realizadas.</p>
path_player	ARRAY(STRING)	<p>Array contendo o percurso que o jogador realizou durante a sessão. Neste campo é interessante informar os objetos que o jogador interagiu e entrou em contato. Ao final da sessão esse campo deve conter informações que mostrem com detalhes a trajetoria do jogador na sessão. Ex: ["Plataforma 1","Plataforma 4", "Plataforma móvel 3"] com esse <i>array</i> de 3 strings é possível entender que na sessão o jogador pulou na plataforma 1, depois plataforma 4 e por ultimo na plataforma móvel 3.</p>

Este é um **campo muito importante** pois este campo que vai determinar a performance do jogador dentro de uma sessão, fica a critério do desenvolvedor informar qual o valor deste