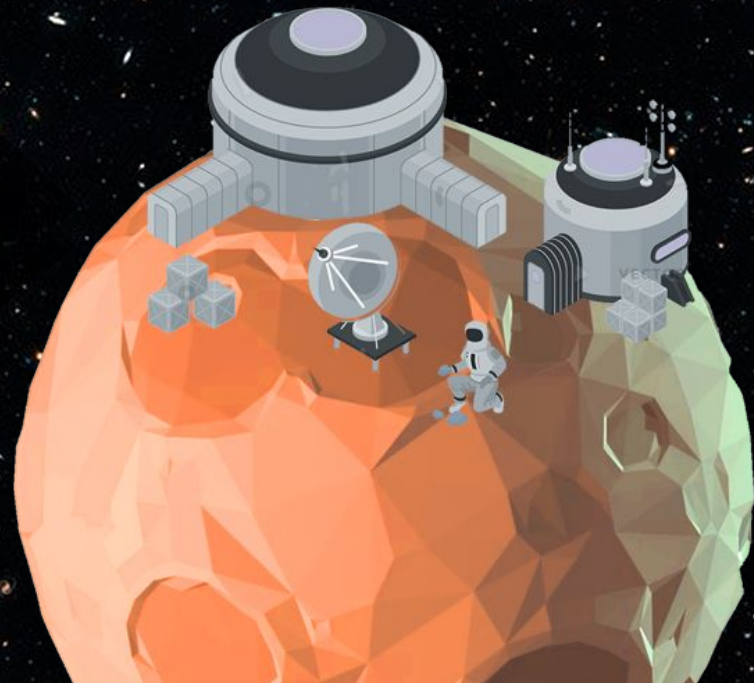


STAR TOWER

Post-Mortem



Equipe

João Vítor Ponte - Programador e Artista

- Implementação do algoritmo de construção de mapas a partir de uma matriz
- Construção da User Interface
- Criação da identidade visual do jogo
- Produção da cutscene

Gabriel Dezan - Programador e Game Designer

- Programação do jogo
- Balanceamento

Sérgio Piza - Música e Efeitos Sonoros

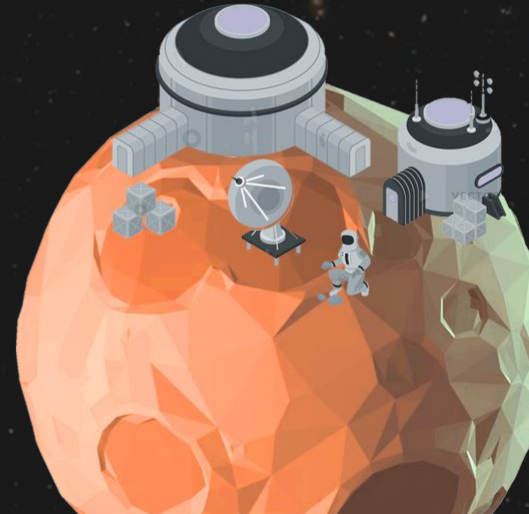
- Produção da música e dos efeitos sonoros do jogo

Angelo Tessaro - História e Documentação

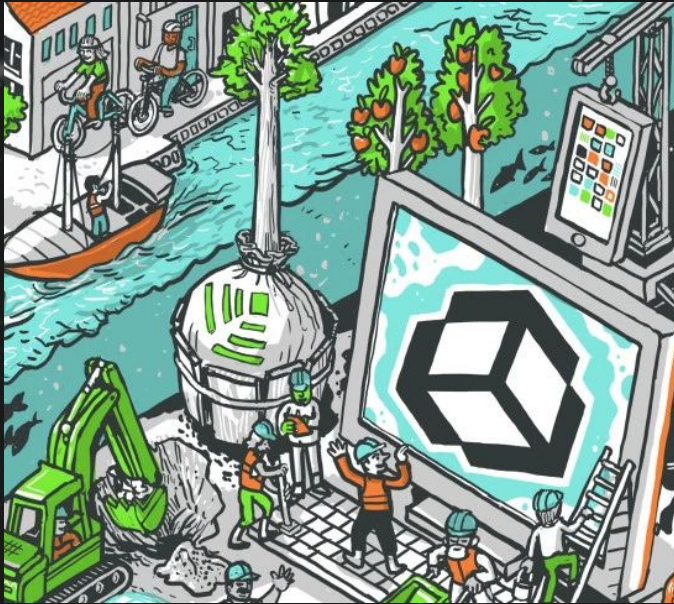
- Construção da História
- Produção One Sheet e Ten-pager



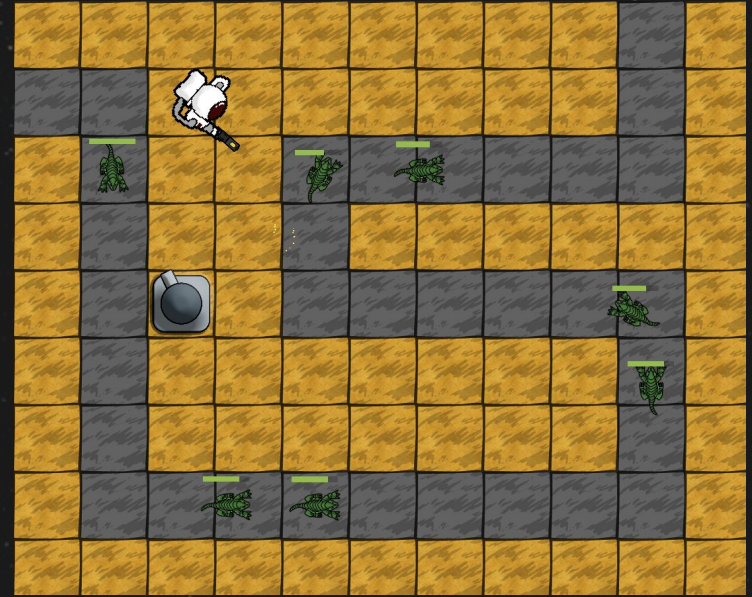
O que deu **certo**?



Aprendizado de Unity



Algoritmo de movimento dos monstros



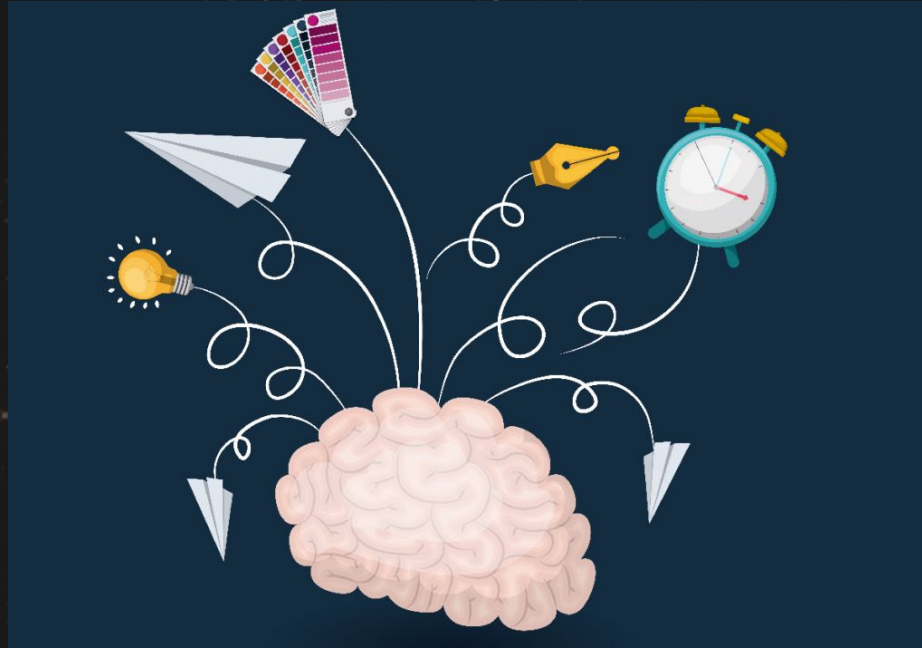
Proposta de interação com o personagem em um TD

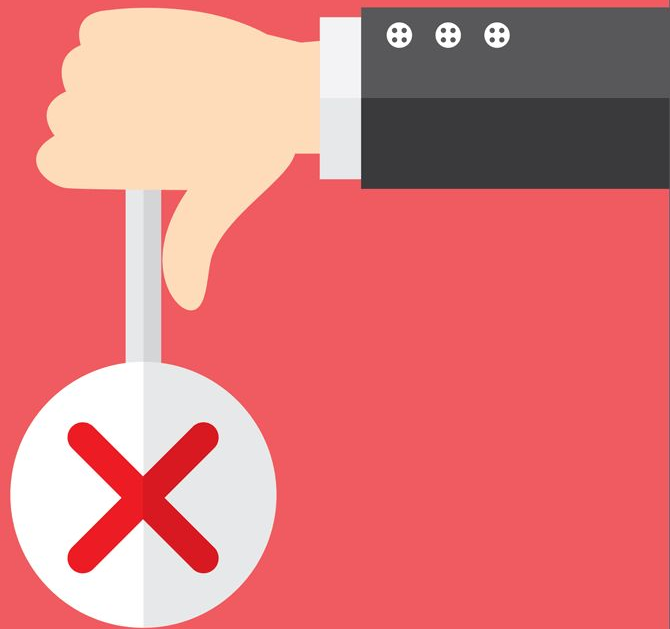


Boa reação do público

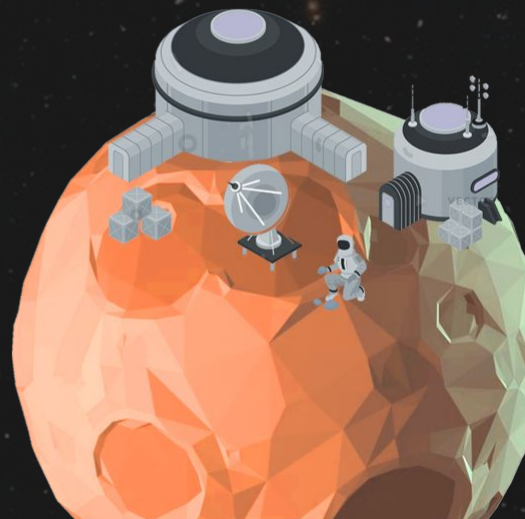


Oportunidade do grupo se expressar criativamente





O que deu **errado?**



Membros Inexperientes



Cronograma



Código pouco escalável

```
317 // Node stopping error is caught below in the select.
318 if err := t.rpcContext.Stopper.RunTask(
319     stream.Context(), "storage.RaftTransport: processing batch",
320     func(ctx context.Context) {
321         t.rpcContext.Stopper.RunWorker(ctx, func(ctx context.Context) {
322             errCh <- func() error {
323                 var stats *raftTransportStats
324                 stream := &LockedRaftMessageResponseStream{MultiRaft_RaftM
325                 for {
326                     batch, err := stream.Recv()
327                     if err != nil {
328                         return err
329                     }
330                     if len(batch.Requests) == 0 {
331                         continue
332                     }
333                     if stats == nil {
334                         stats = t.getStats(batch.Requests[0].FromReplica.NodeID
335                     }
336                     for i := range batch.Requests {
337                         req := &batch.Requests[i]
338                         atomic.AddInt64(&stats.serverRecv, 1)
339                         handleRaftRequest(ctx, req, stream); pErr
340                         stats.serverSent, 1)
341                         response(req, pErr
```

Conclusão

- **Muito aprendido**
- **Boa experiência com a disciplina**
- **Resultado satisfatório**