

Q*bert Universe

Antônio Vinicius de Moura Rodrigues
Edgar Sampaio de Barros

Introdução aos Sistemas Computacionais - Prof. Dr. Marcus Vinicius Lamar
Universidade de Brasília - UnB, Dept. De Ciência da Computação, Brasil

Resumo

No presente artigo iremos abordar como foi pensado e elaborado o projeto prático da disciplina Introdução aos Sistemas Computacionais, tal projeto trata-se do desenvolvimento de uma releitura do jogo de arcade Q*bert, para criar essa nova versão foi utilizado o simulador RARS e suas ferramentas, tais como BITMAP Display, Keyboard and Display MMIO Simulator e sua interface de áudio, com a arquitetura RISC-V.

O jogo original de 1982 Q*bert será analisado detalhadamente no decorrer do artigo, juntamente com a metodologia utilizada para a organização e desenvolvimento do projeto e as dificuldades encontradas ao decorrer progresso.

1. Introdução

Este artigo tem como principal objetivo relatar a experiência do desenvolvimento de uma nova versão do jogo Q*bert utilizando o simulador RARS com a arquitetura RISC-V. A ideia de desenvolver tal uma nova versão do jogo foi proposta como projeto prático da disciplina Introdução aos Sistemas Computacionais ministrada no segundo semestre de 2019 pelo professor Marcus Vinicius Lamar na Universidade de Brasília para calouros do curso de Bacharel em Ciência da Computação.

De acordo com *wikipedia*, Q*bert é um jogo é de plataforma isométrico com elementos de

quebra-cabeça criado pela produtora Gottlieb em 1982 para o console Atari (Figura 1). O jogador controla o personagem-título com o objetivo de alterar a cor de cada bloco da pirâmide, evitando obstáculos e inimigos.

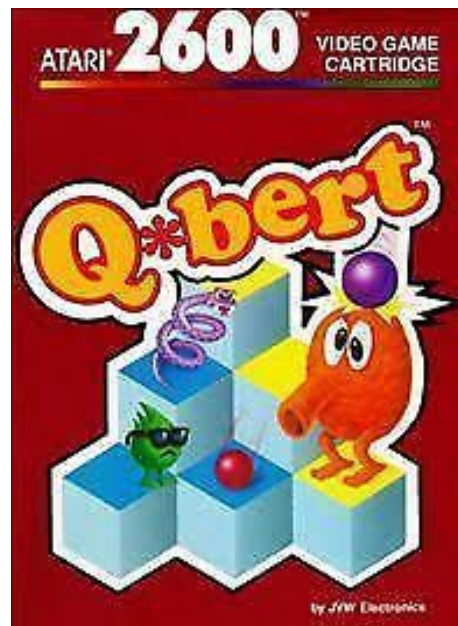


Imagem 1: capa do jogo original

2. Metodologia

O projeto foi desenvolvido utilizando uma versão customizada pelo professor e pelos monitores da disciplina do simulador RARS com a arquitetura de computadores RISC-V. Tal arquitetura é um conjunto de instruções livre que pode ser utilizado por qualquer empresa ou pessoa, facilitando a criação e implementação de chips e softwares para diversas finalidades. Segundo o site oficial da RISC-V, ela foi criada pelo professor Krste Asanović e os alunos de pós-graduação

Yunsup Lee e Andrew Waterman que iniciaram as as instruções RISC-V em maio de 2010.

Como inspiração para a criação do jogo, nas especificações do trabalho foi nos dado um link para uma versão flash do Q*bert (<http://www.q-bert.com/games/qbert.htm>) que deveríamos usar como referência base de mecânicas, sons e layouts básicos do jogo.

Para o desenvolvimento de nossa nova versão do jogo optamos por escolher um tema ambientado no espaço. Para isso criamos uma nova logo (Figura 1) e novas artes para as fases com o tema espacial.



Figura 1: logo Q*bert Universe

O nosso primeiro passo para o desenvolvimento do projeto foi fazer uma leitura crítica da especificação do trabalho fornecida pelo professor, com isso foi possível ter uma ideia geral de como deveria ser a nossa versão do jogo e dividir o desenvolvimento em grandes áreas, sendo elas: arte, som, mecânica e relatório final, cada uma com diversas metas e objetivos gerais a serem cumpridos.

Para auxiliar na organização de referências e metas nos fizemos uso da plataforma online *Trello* (Figura 2) e para o gerenciamento dos arquivos e versões do projeto usamos o *GitHub*.

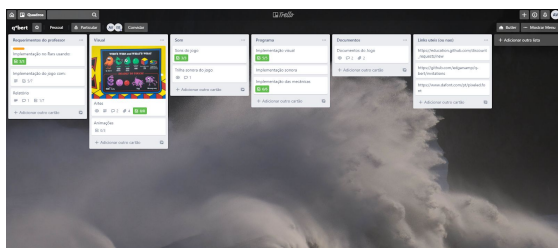


Figura 2: Ambiente Trello

O nosso próximo passo foi analisar os códigos fornecidos pelo professor e seus monitores, nesses códigos continham exemplos de como utilizar as ferramentas de imagens, sons e fazer a leitura de dados do teclado do RARS.

Após a análise dos códigos de exemplo nós os adaptamos para o nosso projeto, criamos um fundo estrelado, um modelo simples do protagonista e o tabuleiro do jogo. Depois de muitas dificuldades conseguimos fazer com que o personagem andasse pelo tabuleiro, para isso criamos um sistema de variáveis onde é armazenado a posição do jogador em relação ao sistema de referência do tabuleiro (figura 3), a linha e posição na linha em que o jogador se encontra.



Figura 3: Sistemas de referência usado

Como o jogo original foi desenvolvido para o console Atari e a nossa versão deveria funcionar para computador tivemos que adaptar o sistema de movimento, no jogo o personagem Q*bert se desloca para suas diagonais, então para facilitar o entendimento do jogador em relação às mecânicas de movimento foram escolhidas as teclas 'Q', 'E', 'A' e 'D' para realizar o deslocamento nas direções esquerda-cima, direita-cima, esquerda-baixo e direita-baixo, respectivamente.

Depois de finalizar o movimento do Q*bert, nos definimos as condições de morte do jogador, caso ele tente andar para fora do tabuleiro é exibido uma mensagem e ele não realiza o movimento.

Com a mecânica de movimento finalizada, o próximo passo foi fazer com que os blocos em que o Q*bert passe devem ter a sua cor alterada e caso o jogador tivesse passado por todos os blocos é exibido uma mensagem dizendo que a fase foi concluída.

Após isso adicionamos um som sempre que o jogador realiza um movimento e criamos duas novas fases (Figura 4 e 5) para o jogo.



Figura 4: fase 1



Figura 5: Fase 2

Como não conseguimos implementar nenhum inimigo em nossa versão do Q*bert, optamos por deixar as pontuações fixas pois o único modo de ganhar é passando de fase.

Por fim, criamos o menu inicial (figura 6) com telas de opções de novo jogo, ajuda e créditos. Além de uma página para a tela de ajuda (figura 7) e créditos (figura 8). Na tela de créditos criamos uma pixel arte do Q*bert com características de cada integrante do grupo de desenvolvimento.



Figura 6: Menu inicial



Figura 7: Tela de ajuda



Figura 8: Tela de créditos

Durante o desenvolvimento do projeto nós nos deparamos com diversas dificuldades, foram elas:

Dificuldade 1

Entender e estudar o Assembly, pois diferente das demais linguagens estudadas pelos membros que compõem nosso grupo, por

exemplo a linguagem C ou Python, ela é uma linguagem de baixo nível, o que dificulta o entendimento do código.

Dificuldade 2

Mais uma dificuldade em relação a linguagem Assembly, para a criação de variáveis é utilizado um sistema diferente, que faz o uso de ponteiros e gerenciamento direto na memória, tal sistema gerou algumas complicações no início do desenvolvimento do nosso projeto.

Dificuldade 3

Utilizar e dominar o simulador RARS nos trouxe diversos desafios, sua interface é pouco intuitiva e durante vários momentos ocorreu do simulador travar e fechar, perdendo todo o progresso feito que não estava salvo.

3. Conclusão

Desde o início, o desenvolvimento desse projeto foi um grande desafio para todo o grupo, apesar das dificuldades encontradas pelo grupo com a divisão de trabalhos e com metas bem estabelecidas foi possível chegar a um ponto satisfatório do desenvolvimento.

A linguagem Assembly foi no início uma grande dificuldade devido a fatores citados no corpo deste trabalho, porém com o desenvolvimento do projeto conseguimos nos adaptar e entender melhor a linguagem e como utilizar suas ferramentas.

Não conseguimos cumprir todas as metas, ainda falta alguns detalhes como a implementação de inimigos e atualizar o sistema de pontos, porém estamos satisfeitos com o trabalho feito até agora e confiantes para continuar o desenvolvimento do projeto.

Referencias:

wikipedia. “Q*bert”. Disponível em: https://en.wikipedia.org/wiki/Q*bert/. Último acesso em: 10 dezembro.2019, 20:34.

riscv. “RISC-V Origin”. Disponível em: <https://riscv.org/risc-v-history/#origin/>.

Último acesso em: 10 dezembro.2019, 20:54.