

4

Resultados

Nesta seção, analisaremos o funcionamento do módulo cinegrafista proposto. Foram realizados dois testes para a avaliação do módulo cinegrafista. O primeiro corresponde a uma análise do custo computacional de cada uma das restrições, assim como uma análise da variação do custo computacional total referente à adição de restrições no sistema. O segundo teste consiste em verificar se o módulo cinegrafista é capaz de posicionar a câmera de maneira satisfatória e se as restrições implementadas conseguem criar, através de seus agrupamentos, os movimentos de câmera esperados. A seguir, é explicado como cada um desses testes foi desenvolvido.

4.1

Análise de desempenho

O cálculo do custo computacional do módulo cinegrafista não pode ser feito a partir de uma configuração específica. Como o módulo é configurado a partir de uma composição qualquer de restrições, primeiramente propomos uma análise de desempenho do pior caso para cada restrição apresentada. O pior caso de uma restrição ocorre quando a câmera não atende às definições da restrição. Por exemplo, o pior caso da restrição de orientação ocorre quando a câmera não atende a nenhum dos limites dos três eixos de rotação definidos pela restrição, forçando a restrição a atuar.

A Figura 4.1 apresenta um gráfico do custo computacional em milissegundos de cada restrição. Para a obtenção dos valores, foi utilizado um computador Pentium IV 2.4Ghz com 512Mb de memória RAM e retirada a média do tempo gasto para a restrição atuar dez mil vezes em seu pior caso.

A partir do custo por iteração de cada restrição é possível avaliar um custo aproximado do módulo como um todo. A função de custo se baseia no número de iterações do sistema ($nItr$), no custo das restrições adicionadas (r_i) e no custo das dez restrições de barras rígidas que pertencem ao modelo

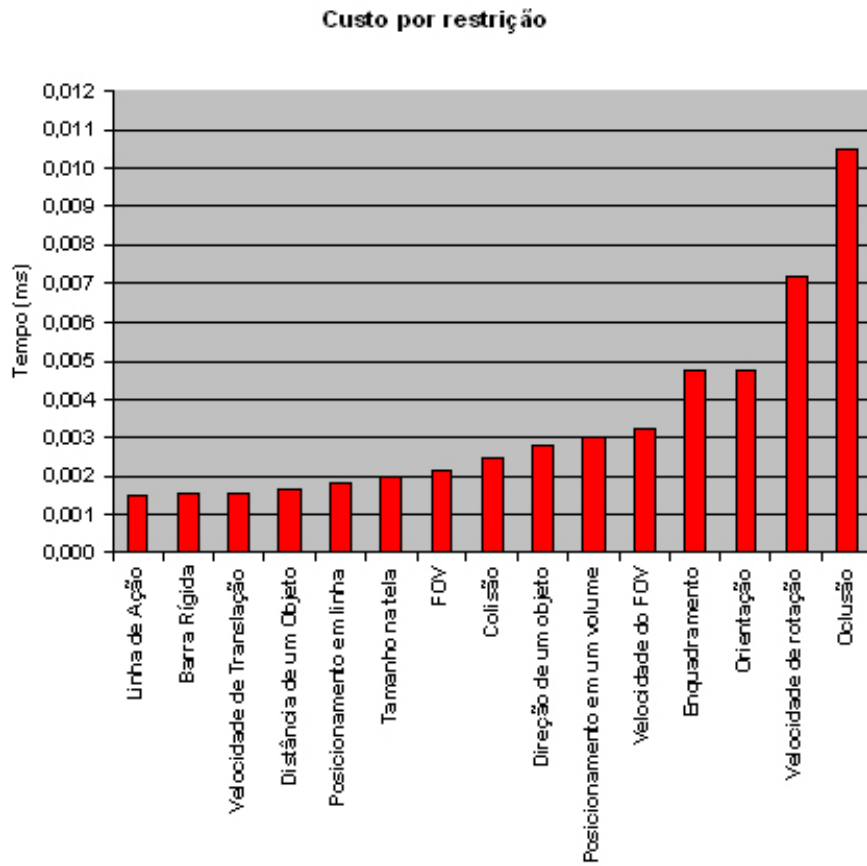


Figura 4.1: Custo individual de cada restrição

físico (*stick*). Como os custos utilizados correspondem ao custo de pior caso, a função do custo do módulo também se refere ao seu pior caso. A equação 4-1 permite calcular o custo computacional total do módulo com diferentes restrições.

$$custo = nItr \cdot (10 \cdot stick + \sum_{i=1}^k r_i) \quad (4-1)$$

Com o objetivo de validar a fórmula acima, foi realizado um experimento comparando o custo calculado com o custo real. O custo real foi calculado posicionando a câmera em um estado no qual nenhuma restrição estivesse sendo atendida, e a partir daí foi calculado o custo computacional da execução das $nItr$ iterações. A Figura 4.2 permite observar que o custo real acompanha linearmente o crescimento do custo calculado à medida que novas restrições são adicionadas ao módulo. Dessa forma, é possível fazer uma aproximação do custo computacional real do módulo a partir do conhecimento dos custos individuais das restrições e do número total de iterações do sistema.

Este resultado demonstra que as restrições são tratadas de forma independente umas das outras. Isto permite a especificação de configurações de movimento de câmera sofisticadas através da composição de restrições simples, com um custo computacional esperado conhecido.

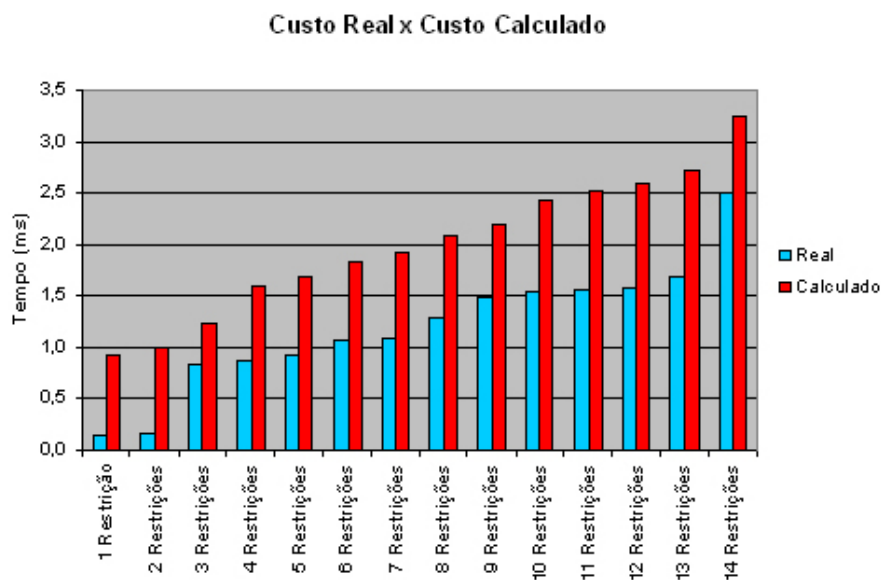


Figura 4.2: Comparação entre custo real e custo calculado

A partir dos desempenhos constatados, podemos concluir que a maioria das restrições apresenta baixo custo computacional. Entretanto, deve-se perceber que o custo da restrição de colisão e o da restrição de visibilidade dependem fundamentalmente da complexidade da cena, isto é, quanto mais complexa a descrição da cena, maior será o custo computacional dessas restrições. Uma cena formada por uma esfera foi utilizada na análise de desempenho da restrição de colisão. A cena utilizada para medir o desempenho da restrição de visibilidade correspondia a uma esfera impedindo a visibilidade da esfera envolvente do objeto de interesse.

4.2

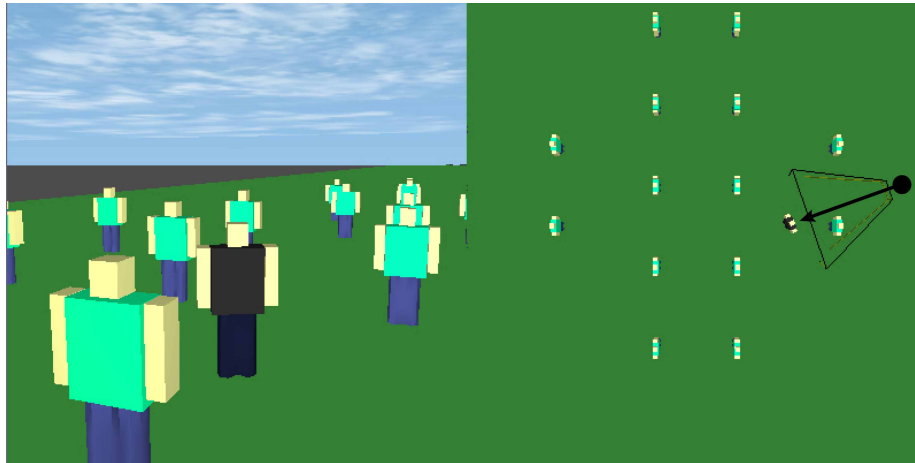
Exemplos de configurações de câmera

Foram criados modelos de movimentação de câmera utilizando apenas as restrições sem a utilização de forças externas com o objetivo de demonstrar que a composição de restrições simples é capaz de realizar configurações de câmera usualmente encontradas na cinematografia e nos jogos eletrônicos.

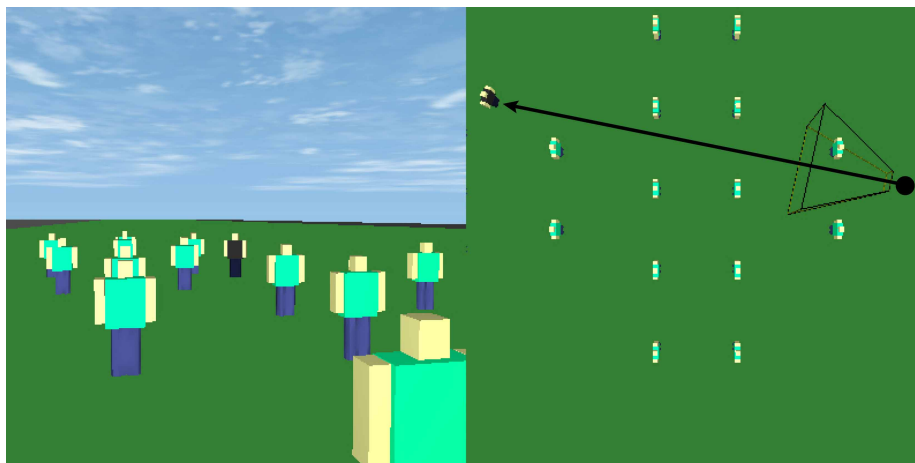
Câmera fixa: Este experimento consiste em posicionar a câmera em um local fixo na cena e enquadrar um determinado objeto no centro da

tela. A ação de movimento do objeto deve provocar uma reação da câmera que altere apenas sua orientação, de modo que o objeto esteja sempre no centro da tela.

Para criar esta configuração foi adicionada uma restrição de *Posicionamento em um volume* com prioridade 1,0 que fixa a câmera em uma posição, uma restrição de *Enquadramento* com prioridade 0,05 que força o objeto a aparecer sempre no meio da tela, e uma restrição de *Orientação* com prioridade 0,9 que fixa o *up* da câmera. O uso de uma baixa prioridade para a restrição de enquadramento permitiu que a câmera assumisse um movimento mais suave, criando um atraso na rotação da câmera em relação à movimentação do objeto. A Figura 4.3 mostra a configuração da câmera em duas posições diferentes do personagem.



4.3(a):



4.3(b):

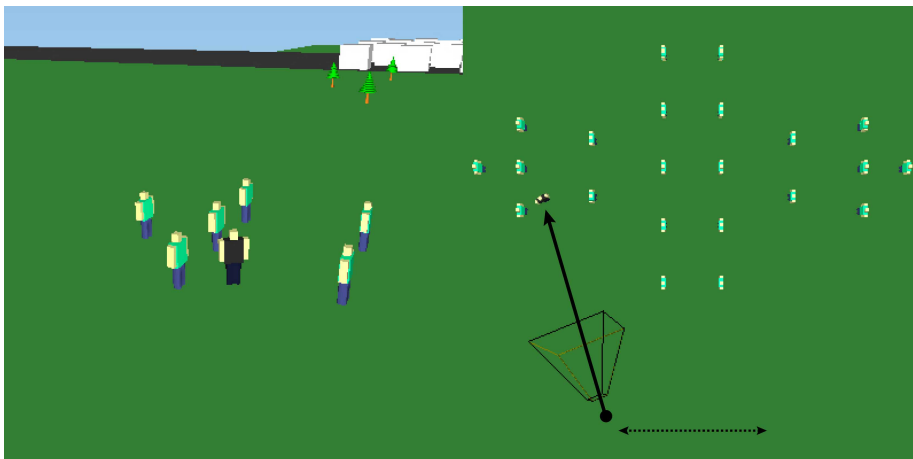
Figura 4.3: Câmera fixa: nas Figuras (a) e (b) a câmera se mantém na mesma posição alterando apenas sua orientação para enquadrar o personagem no centro da tela.

Câmera sobre trilho: Esta configuração posiciona a câmera sobre uma linha, procurando manter uma distância mínima de um objeto que deve sempre ser enquadrado no centro da tela. Movimentos como este são utilizados quando se procura seguir um determinado personagem e, em jogos eletrônicos, a câmera principal de jogos esportivos, como futebol, possui comportamento semelhante.

Este movimento foi realizado adicionando uma restrição de *Posicionamento em segmento de reta* (prioridade 0,5) além de restrições quanto à *Velocidade de translação*(0,9), *Orientação*(0,9), *Velocidade de rotação*(0,2), *Enquadramento*(0,1), *Visibilidade*(0,5) e *Distância de um objeto*(0,2). A Figura 4.4 ilustra o posicionamento da câmera em duas posições diferente do personagem.



4.4(a):

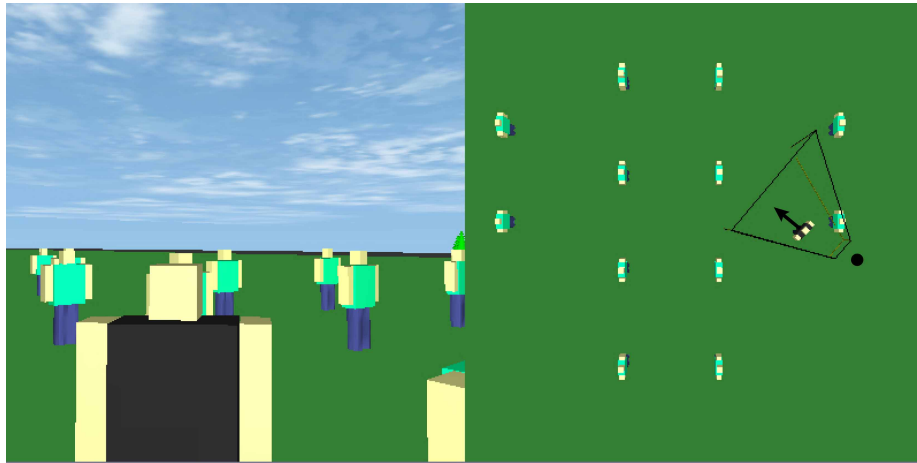


4.4(b):

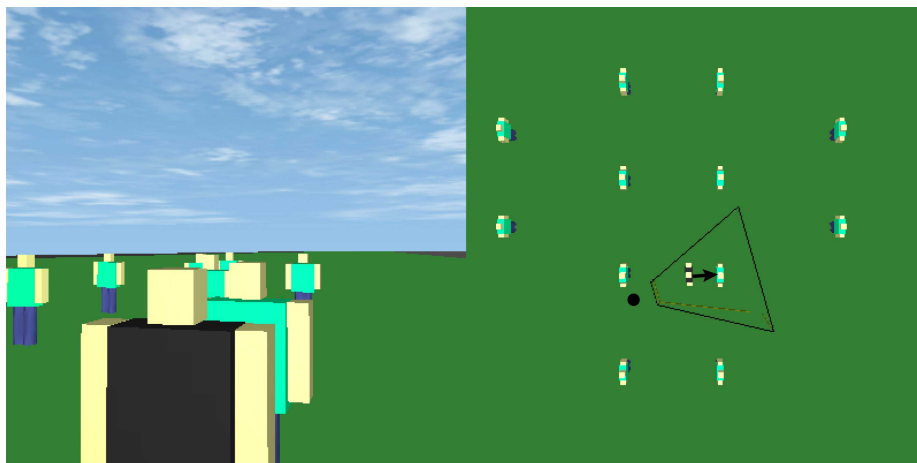
Figura 4.4: Câmera sobre trilho: nas Figuras (a) e (b) a câmera procura se manter sobre o trilho enquadrando o personagem no centro da tela. Na Figura (b) as restrições de *distância de um objeto* e *posicionamento em segmento de reta* não conseguem ser atendidas simultaneamente, como o resultado a câmera assume uma posição intermediária.

Câmera sobre os ombros: A câmera sobre os ombros é muito utilizada em jogos com visualização de terceira pessoa. Consiste em posicionar a câmera sempre atrás de um objeto, deixando-o levemente deslocado do centro da tela.

Esta configuração foi conseguida através da adição de restrições de *Orientação*(0,9), *Distância de um objeto*(0,1), *Direção de um objeto*(0,01), *Enquadramento*(0,5) e *Colisão*(0,05). O uso de uma esfera envolvente na restrição de colisão cria uma área onde a esfera colide com a cena mas o plano *znear* de visualização não é prejudicado. Isto permite que valores mais baixos de prioridade sejam utilizados na restrição de colisão, criando uma desaceleração da câmera à medida que esta se aproxima de um obstáculo. A Figura 4.5 ilustra o posicionamento de uma câmera localizada sobre os ombros de um objeto.



4.5(a):

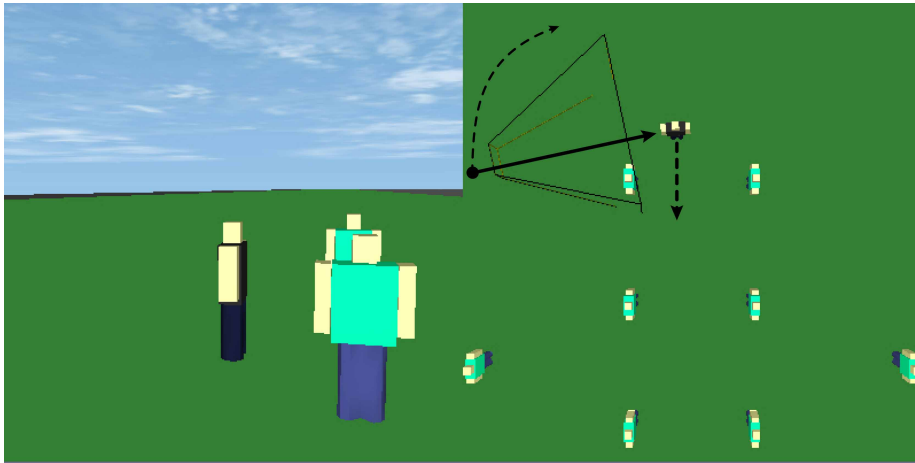


4.5(b):

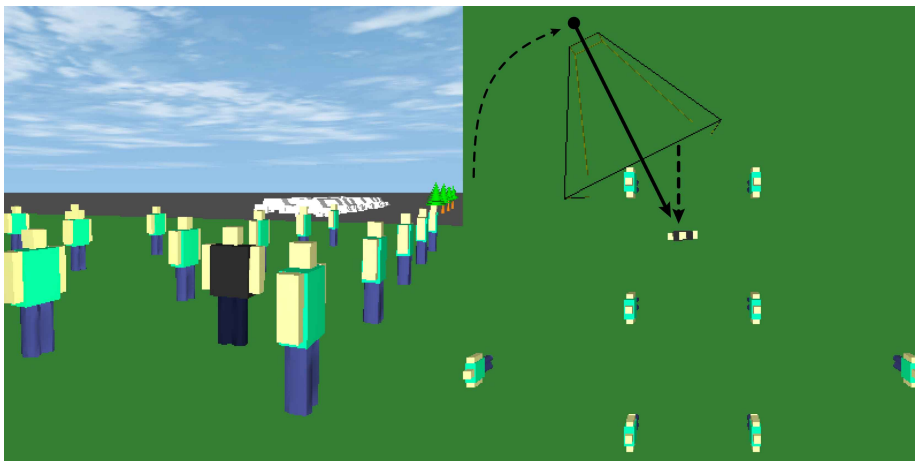
Figura 4.5: Câmera sobre os ombros: nas Figuras (a) e (b) a câmera assume uma posição em relação a direção que o personagem está voltado. O enquadramento do personagem é levemente deslocado do centro da tela.

Câmera seguindo o alvo: Similar à configuração *sobre os ombros*, este movimento de câmera procura seguir um objeto de interesse, mas sem a preocupação de se posicionar atrás do objeto. No entanto, como a câmera é posicionada relativamente afastada do objeto de interesse, a restrição de visibilidade passa a ter grande importância em sua movimentação

A configuração deste movimento foi conseguida através da adição das restrições de *Orientação*(0,7), *Distância de um objeto*(0,1), *Direção de um objeto*(0,9) forçando a câmera a atuar em uma altura específica em relação ao objeto de interesse, *Enquadramento*(0,5), *Colisão*(0,05) e *Visibilidade*(0,01). A Figura 4.6 ilustra o posicionamento de uma câmera que segue um alvo se movimentando em decorrência da restrição de visibilidade.



4.6(a):

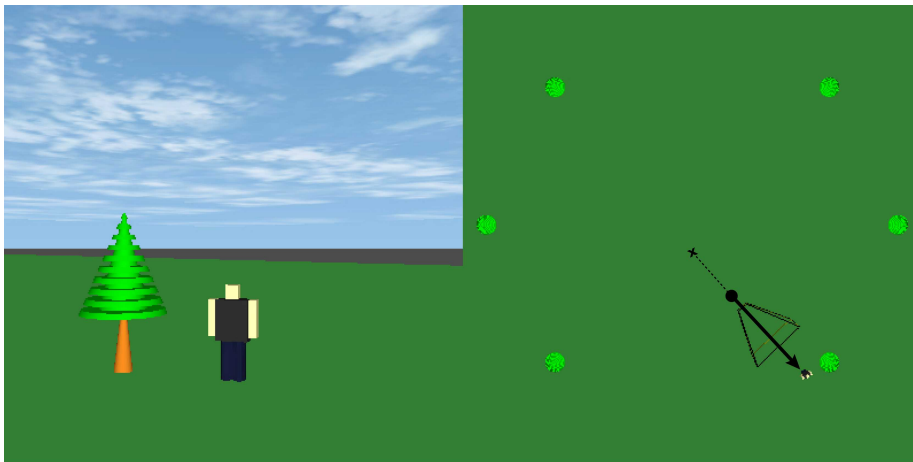


4.6(b):

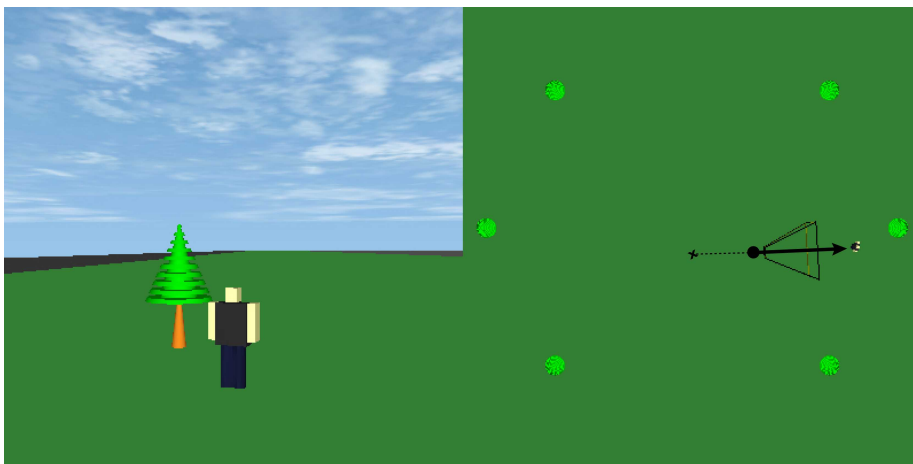
Figura 4.6: Câmera seguindo o alvo: o movimento do personagem gera uma movimentação de câmera com o objetivo de evitar que o personagem fique ocluído.

Câmera em grua: Consiste em simular o posicionamento de uma câmera como se estivesse em cima de uma grua, é uma espécie de guindaste com uma câmera fixada na extremidade. A grua possui uma haste de determinado comprimento, uma de suas extremidades é fixa, enquanto a câmera fica posicionada na outra extremidade da haste. A câmera procura sempre enquadrar um objeto no centro da tela, mantendo-o sempre dentro de uma margem de distância.

Para simular este movimento é adicionado um objeto de referência no qual se localiza a base da grua, e uma restrição de *Distância de um objeto*(0,8) ligada a esse objeto de referência define a distância entre a base e a câmera. Outras restrições ligadas ao objeto de interesse criam o movimento necessário: *Orientação*(0,9), *Distância de um objeto*(0,1), *Enquadramento*(0,9) e *Velocidade de translação*(0,9). A Figura 4.7 ilustra o funcionamento da câmera em grua.



4.7(a):

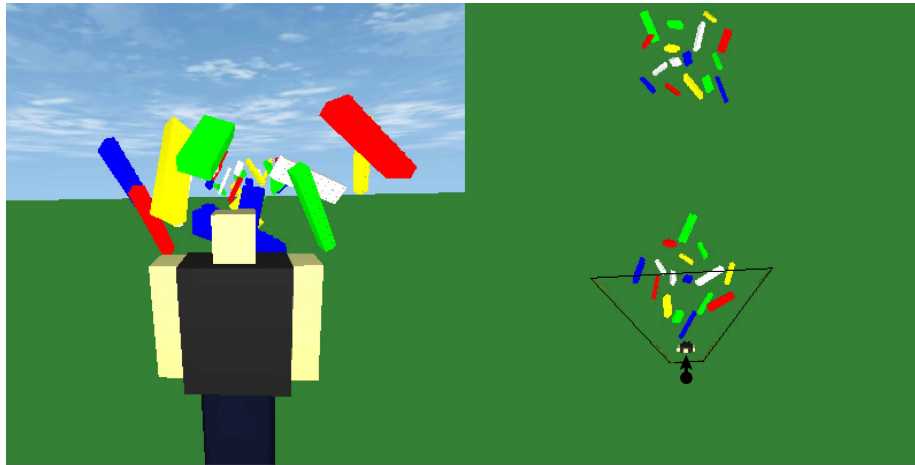


4.7(b):

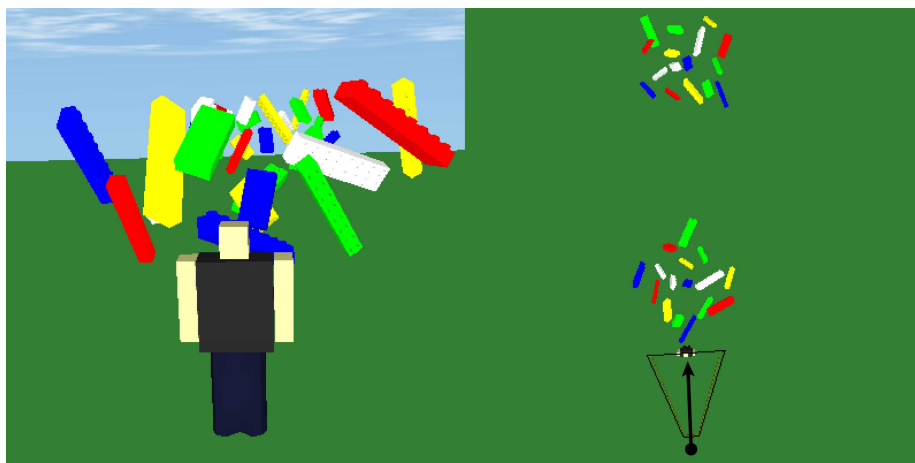
Figura 4.7: Câmera em grua: nas Figuras (a) e (b) a câmera procura manter uma distância tanto de uma base quanto do personagem.

Tomada de câmera “vertigo”: Este movimento de câmera criado por Alfred Hitchcock consiste em alterar o campo de visão da câmera enquanto se altera a distância ao objeto de interesse, em uma relação que mantenha o objeto sempre do mesmo tamanho na tela. Dessa forma, toda a cena é distorcida exceto o objeto de interesse.

A configuração deste efeito pode ser conseguida através da adição das restrições de *Enquadramento*(0,9), que posiciona o objeto de interesse no centro da tela, *Velocidade do FOV*(0,5), que força o campo de visão a ser alterado com o tempo, *Tamanho do objeto*(0,1), que fixa um tamanho e obriga a câmera a se deslocar na proporção correta, e *Distância do objeto*(0,9), responsável por definir um limite quanto à deformação que a câmera pode gerar. A Figura 4.8 apresenta duas tomadas do efeito vertigo.



4.8(a):



4.8(b):

Figura 4.8: Tomada de câmera “vertigo”: a variação do fov e da distância da câmera ao personagem distorce toda a cena exceto o personagem.