Interactive Wind

Diogo Gonçalves (A81860)
Miguel Carvalho (A81909)

22 de Julho, 2020

Neste artigo apresentamos um modo de simular vento e a sua interação com o ambiente que o rodeia.

Para a simulação do vento implementamos uma simplificação da equação de *Navier-Stokes* que fornece um resultado credível sem custar muito a nível de desempenho.

Relativamente ao ambiente que interage com o vento, simulamos lâminas de relva que sofrem modificações consoante as forças que lhes são aplicadas (*i.e.*, força da gravidade e do vento).

Além disso, apresentamos também uma forma simples de simular o vento a contornar obstáculos numa cena.

1 VENTO

O vento como um fenómeno físico pode ser representado através de mecânica de fluídos. Para tal, utilizamos as equações de *Navier-Stokes*. Estas consistem num conjunto de equações diferenciais que descrevem o movimento de substâncias fluídas viscosas. No caso particular do vento e de forma a simplificar a aplicação das equações iremos assumir alguns pressupostos importantes, tais como, o vento é um fluído inviscível, irrotacional e incompressível. Esta é uma modelação razoável para modelos com velocidades "normais"e sem turbulência.

1.1 Tipos de vento

Os diferentes tipos de ventos que serão apresentados tem todos o mesmo requisito de satisfazer a equação de *Navier-Stokes*. Esta característica permitirá que os diferentes tipos de ventos possam ser combinados.

1.1.1 ***Sink*** . Neste efeito existe um ponto central onde todas as velocidades convergem na direção desse determinado ponto. É importante referir que para uma determinada distância independentemente da posição os pontos têm todos a mesma velocidade. Este efeito está ilustrado na figura 1.

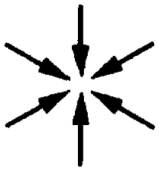


Fig. 1. Efeito Sink

$$\phi = \frac{a}{2\pi} ln(r); \quad v_r = \frac{a}{2\pi r}; \quad v_\theta = 0; \quad v_z = 0.$$

Para perceber exatamente como funciona o *Sink* iremos analisar mais concretamente as equações resultantes das equações *Navier-Stokes*. As equações em 1.1.1 estão representadas em coordenadas cilíndricas. É possível observar que a existem apenas 2 variáveis: o raio r e a força a.

O raio corresponde à distância entre um determinado ponto ao ponto central. Como verificamos nas equações 1.1.1, podemos concluir que o efeito é inversamente proporcional ao raio, ou seja, verificamos que quanto mais um um ponto dista do centro menos força sentirá e que os pontos à mesma distância sentirão uma força igual (em coordenadas cilíndricas).

A outra variável é o a que representa a força na origem do *sink*. Para que os pontos se sintam atraídos para a origem, esta componente tem que ser negativa.

1.1.2 ***Source*** . O efeito *Source* é o oposto do efeito *Sink*. Na figura 2 conseguimos observar que, neste caso em concreto, as forças em vez de convergir de um ponto central irão divergir.

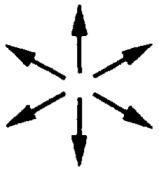


Fig. 2. Efeito source

$$\phi = \frac{a}{2\pi} ln(r); \quad v_r = \frac{a}{2\pi r}; \quad v_\theta = 0; \quad v_z = 0.$$

As equações para este efeito (*i.e.*, 1.1.2) são exatamente iguais às equações para o efeito *Sink* (*i.e.*, 1.1.1), o que confirma a premissa que os efeitos são opostos. Neste caso, a única coisa que difere é a variavel a que tem obrigatoriamente de ser um valor positivo, de modo a que os pontos sintam uma força divergente.

1.1.3 ***Vortex*** . O último tipo de vento que ainda depende das equações *Navier-Stokes* é o *Vortex*. Na figura 3 dá para percecionar que terá que ser feita abordagem diferente aos efeitos anteriormente descritos.

Fig. 3. Efeito vortex

Neste caso, as velocidades irão manter a mesma distância mas curvar de acordo com o sentido de uma força.

$$\phi = \frac{b}{2\pi}\theta; \qquad v_r = 0; \qquad v_\theta = \frac{b}{2\pi r}; \qquad v_z = 0.$$

Nas equações 1.1.3 é possível analisar que a componente v_r tem valor 0, no entanto, a componente v_θ da velocidade deixa de ser 0. Isto significa que de acordo com as coordenadas cilíndricas a variação dessa componente resultará numa espécie de circulo à volta de um ponto central.

1.1.4 ***Uniforme*** . Por fim, falta o tipo de vento Uniforme. Este tipo de vento não requer qualquer tipo de auxilio às equações de *Navier-Stokes* uma vez que só é necessário definir uma direção , uma intensidade e um pouco de *Perlin noise* de forma a tornar o vento mais realista. Apesar disto continua a respeitar os requisitos das equações de *Navier-Stokes*

1.2 Adição de vento

Uma vez que todos os tipos de ventos anteriormente descritos consistem de sistemas lineares e obedecem ao teorema de Navier-Stokes, todos os efeitos podem ser somados de forma a criar novos efeitos:

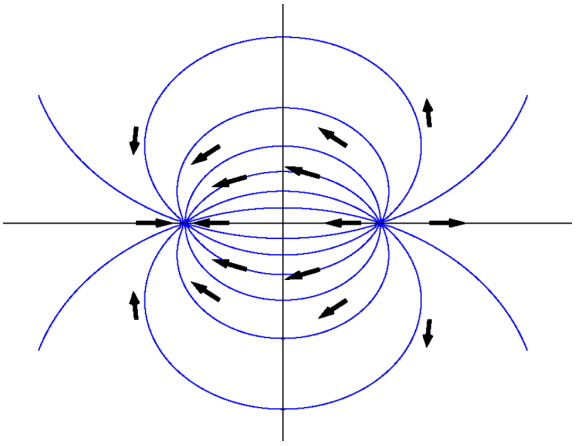


Fig. 4. Um *source* e um *sink*

$$v_{total} = v_{vortex}(x,y,z) + v_{sink}(x,y,z) + v_{source}(x,y,z) + v_{uniforme}(x,y,z) \quad (1)$$

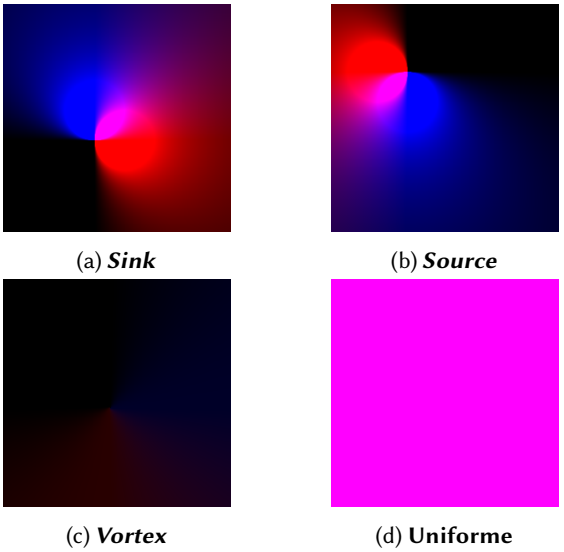
Na equação 1 conseguimos observar que a soma linear de todas as velocidades resulta na velocidade final para um determinado ponto.

Esta combinação permite criar resultados interessantes e fisicamente corretos. Na imagem 4 conseguimos observar o resultado da colocação em paralelo de um *Sink* e uma *Source*. As linhas de campo observadas são resultantes da combinação linear de ambos os pontos.

1.3 Implementação - Nau3D

Uma vez percebido o funcionamento e a abordagem de cada um dos tipos de ventos está na altura de começar a perceber como integrar esta abordagem no *engine* Nau3D. A melhor forma de conseguir obter um bom desempenho em tempo real da simulação de vento será através da utilização de *shaders* tal como o compute. Nesta implementação em concreto serão feitos todos os cálculos num *compute shader* e gravados numa textura para a posteriori esta ser acedida no pipeline.

1.3.1 ***Tipos de vento*** . Com o intuito de conseguir observar corretamente se o que foi apresentado anteriormente está ou não correto serão exibidas as texturas guardadas para os diferentes tipos de vento.



(a) ***Sink*** (b) ***Source*** (c) ***Vortex*** (d) **Uniforme**

Fig. 5. Vários tipos de vento representados em texturas.

2 AMBIENTE

De maneira a que a presença do vento seja sentida numa cena, é preciso que nela existam objectos que

interajam com o mesmo. Com isso em mente, decidimos simular lâminas de relva que se movessem com o vento de uma maneira realista.

Decidimos que para o nosso projeto queríamos que as lâminas de relva fossem "procedurally generated", sendo estas controladas por vários tipos de parâmetros (*e.g.*, comprimento ou rigidez). Desse modo, inspirando-nos no trabalho feito em [1] e [3], definimos as lâminas de relva com recurso a curvas de *Bezier*. Em [3] é usado um modelo pré-definido com os quatro pontos de controlo das curvas já escolhidos, enquanto que em [1] são usados apenas três pontos de controlo, onde o primeiro ponto é a base da lâmina, o segundo é obtido através da aplicação das forças que interagem com a lâmina e o terceiro é obtido com recurso ao primeiro e ao segundo.

Neste projeto, aplicamos preferencialmente as ideias de [1], sendo retiradas algumas propriedades de [4]. Nas próximas secções apresentamos todo o processo de desenho de uma lâmina de relva, sendo este processo depois replicado, com variação nos diversos parâmetros, para gerar um campo de relva.

2.1 Esqueleto

O esqueleto de uma lâmina de relva é inicialmente apenas uma reta, definida por dois pontos, perpendicular ao terreno. Destes dois pontos, são aplicadas apenas transformações ao ponto que não pertence ao terreno.

As primeiras transformações aplicadas são a definição da comprimento, orientação e inclinação da lâmina de relva. Estas, essencialmente, consistem na escolha de um ponto numa hemisfera de raio igual ao comprimento. Todas estas transformações são definidas pelo utilizador, sendo também embutidas de algum ruído (controlável pelo utilizador) para atribuir um ar natural à relva. Na figura 6 podemos observar o esqueleto inicial de uma lâmina e esqueleto transformado com estas operações.

Depois da ponta do esqueleto rígido ter sido definida, começamos a aplicar as forças à lâmina. Começamos pela força da gravidade, sendo esta definida pela soma da gravidade do ambiente e a gravidade frontal. Para definir estas forças, recorremos às equações 4, 5 e 6 de [1] e aplicamos algumas modificações, passando estas a estar definidas por:

$$g_E = m\overrightarrow{g}; \qquad g_F = \frac{1}{4}\|g_E\|f; \qquad g = g_E + g_F,$$

onde m é a massa da lâmina de relva, $\overrightarrow{g}$ é o vetor que representa a gravidade e f é o vetor que representa a direção frontal da lâmina, que é perpendicular à largura da lâmina.

Fig. 6. Aplicação da comprimento, inclinação e rotação.

Após calculada a força gravítica, esta é somada à ponta da lâmina da relva.

Além da força gravítica, é também aplicada a força do vento que é consultada na textura mencionada anteriormente, usando as coordenadas do mundo da ponta rígida da lâmina. No entanto, esta força não é somada diretamente pois esta deve depender do quão dobrada a lâmina está e da sua orientação (e inclinação) em relação à direção do vento. Para ter em conta estes parâmetros, são usadas as equações 7 e 8 do artigo [1].

Para resistir a estas forças, é calculada e adicionada uma terceira força chamada resistência. Esta é a força que contraria as outras forças a ser aplicadas na lâmina e é direcionada para a posição rígida original da ponta da lâmina, antes de lhe serem aplicadas as forças. Desta maneira, esta força cancela todas as outras parcialmente, sendo a percentagem de cancelamento definida pela rigidez da lâmina, que é definida pelo utilizador.

Na figura 7 podemos ver como essas forças são aplicadas ao vértice que representa a ponta da lâmina.

Após todas as forças aplicadas, avançamos para o cálculo do terceiro ponto de controlo da nossa curva de *Bezier*. Para o fazer, escolhemos um ponto colocado entre a base e a posição original da ponta antes de lhe serem aplicadas as forças. No entanto, este ponto não pode ser escolhido arbitrariamente pois este tem influência no comprimento curva gerada. Assim, usamos as equações 12 e 13 em [1] para fazer a escolha do ponto e garantir que o comprimento da lâmina se mantém.

Na figura 8 podemos ver os três pontos de controlo e a curva que os mesmos criam, curva essa que definirá a nossa lâmina de relva.

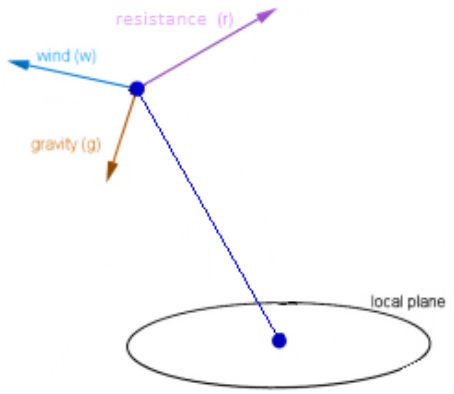


Fig. 7. Aplicação da gravidade, do vento e da recuperação.

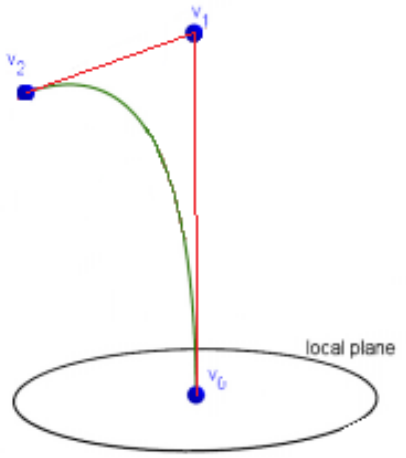


Fig. 8. Cálculo do terceiro ponto de controlo.

2.2 Desenho da Lâmina

Depois do cálculo do esqueleto, é preciso transformar o esqueleto numa lâmina de relva. Para isso, começamos por fazer tesselação na curva de *Bezier* para obter mais pontos e assim diminuir a angularidade do nosso esqueleto. Esta tesselação é controlada pelos níveis da lâmina, que é um parâmetro que o utilizador controla e define o número de subdivisões que existirá na curva.

Tendo a curva sub-dividida, é preciso tornar cada segmento em planos visíveis por uma câmara. Para o efeito é usado o *geometry shader*, que para cada linha correspondente a um segmento, vai criar um retângulo com quatro pontos. Esses pontos são obtidos usando a tangente a cada ponto, ou seja, os pontos obtidos são os pontos à "esquerda"e à "direita"dos pontos do esqueleto, consoante a largura que o utilizador deseja para as lâminas de relva.

Após o passo do *geometry shader*, ficamos com um retângulo curvado graças aos vários sub-retângulos. No entanto, uma lâmina de relva não tem a forma de um retângulo e para solucionar isso e aproximar a forma da relva à realidade são usadas texturas de relva que possuem transparência na área que não corresponde à lâmina. Deste modo, as nossas lâminas terão a forma de lâminas reais e podemos usar vários tipos de textura para desenhar vários tipos de plantas.

Na figura 9 podemos ver as subseções criadas pela tesselação, assim como os pontos que o *geometry shader* cria e também como a textura se aplica à forma retangular.

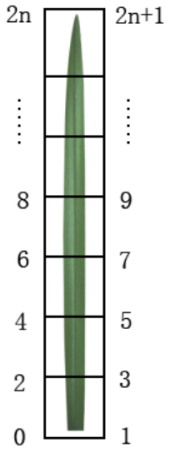


Fig. 9. Tesselação e transformação do esqueleto.

2.3 Resultados

Após a implementação do vento e da relva, foi preciso passar aos ajustes dos parâmetros. Graças à quantidade de parâmetros que é possível ajustar, foi possível criar um campo relvado com um grau de realismo muito aceitável.

Na figura 10 podemos observar o exemplo de campo de relva que usamos nos nossos testes (ainda sem o efeito de vento) e é de notar a variedade que existe neste campo, o que a nosso ver torna a cena muito mais realista.

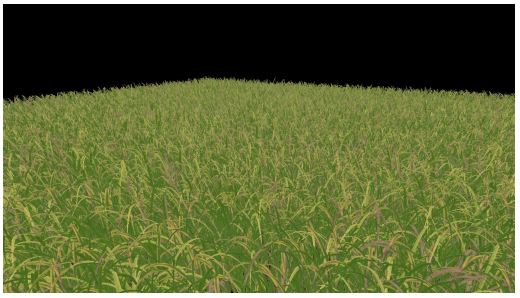
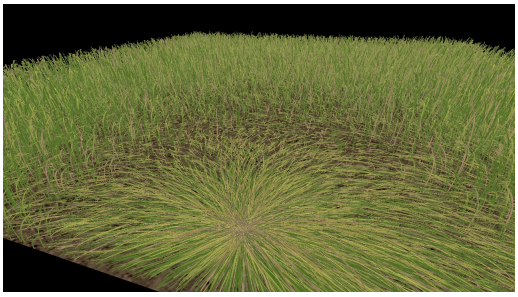


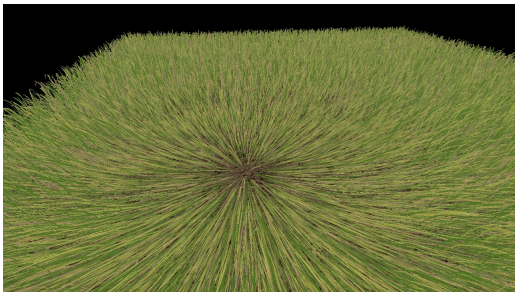
Fig. 10. Campo relvado (sem vento).

No entanto, nós queremos é visualizar a interação do campo relvado quando este se encontra nos

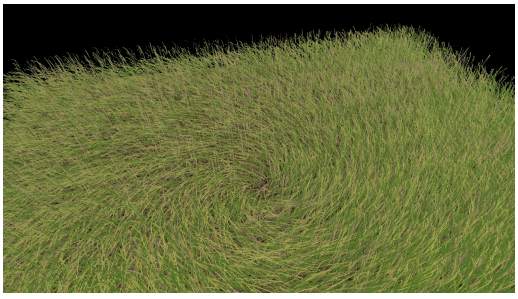
vários campos de vento apresentados acima. Assim, apresentamos *screenshots* de seguida que apresentam a reação do nosso campo, mesmo não sendo possível visualizar o movimento que as lâminas de relva apresentam.



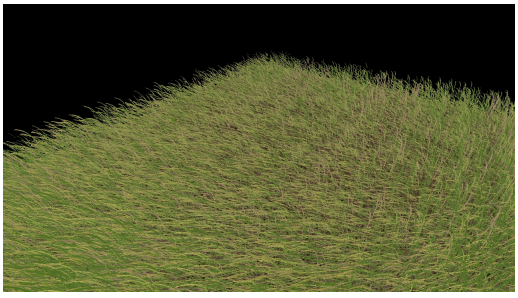
(a) ***Sink***



(b) ***Source***



(c) ***Vortex***



(d) **Uniforme**

Fig. 11. Campo relvado a interagir com vários tipos de vento.

3 OBSTÁCULOS

De modo a melhorar as interações com o vento e de modo a tornar a simulação de vento mais parecida com o mundo real, iremos incluir obstáculos na cena. Uma vez que a ideia do projeto é a interação e não a colisão com obstáculos, poderá acontecer de certas lâminas de relva entrarem para dentro do objeto.

3.1 *Voxelization*

Para proceder ao desvio do vento consoante os objetos, é necessário que o vento conheça a posição dos objetos e as suas caraterísticas. Neste caso, a caraterística que precisamos de conhecer é a normal das faces do objeto. Assim, foi preciso engendrar uma maneira de guardar as normais dos objetos para que na geração do vento fosse possível verificar se o vento deveria ou não sofrer alteração consoante as mesmas.

Essa maneira foi a voxelização, sendo que no nosso caso foi apenas necessária a voxelização 2D. Deste modo, colocamos duas câmaras no centro da cena, com os valores do tipo *near* e *far* ajustados para cobrir a cena, e alinhados com os eixos x e z. De seguida, designamos um passo que faz recurso de uma dessas câmaras (a que melhor visualizar o objeto) para guardar as normais numa textura 2D que partilha a mesma área que a textura do vento.

Na figura 12 podemos visualizar um objeto representado numa textura apenas pelas suas normais.

Fig. 12. Normais dos obstáculos guardadas em textura.

3.2 Interações e efeitos

De forma a representar corretamente a interação do vento com objetos é necessário analisar caso a caso, querendo isto dizer que é necessário criar um caso para cada tipo de vento.

3.2.1 ***Sink***. Na figura 13 é possível verificar que a face do obstáculo se encontra em virada para o *Sink* não exibe qualquer tipo de alteração. Por outro lado, a face de atrás do obstáculo não pode sofrer a mesma força do *Sink* do que as restantes. Para este caso é aplicada uma mistura entre a força perpendicular à normal obtida na voxelização e a força exercida pelo *Sink*. Sendo que quanto mais longe do obstáculo maior será a força relativa da componente *Sink*.

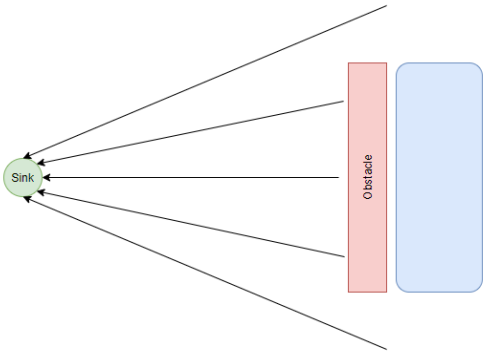


Fig. 13. Campo relvado (sem vento).

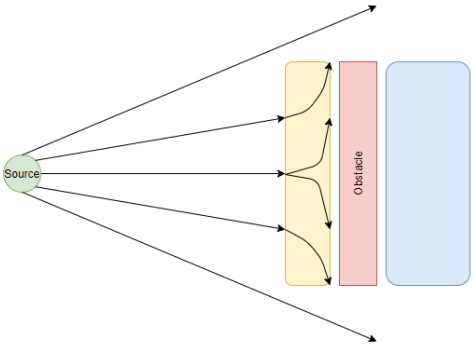


Fig. 14. Campo relvado (sem vento).

3.2.2 ***Source***. A figura 14 expõe o comportamento da interação de uma *Source* com um obstáculo. Neste caso existem 2 momentos.

Numa primeira instância o embate do vento no obstáculo irá criar uma espécie de fenda no meio que resultará em que o vento seja reconduzido na direção da normal obtida a partir da voxelização, tal como exposto na imagem. Para obter este efeito é utilizada a mesma estratégia do *Sink* onde é misturada a perpendicular à normal do obstáculo e a força exercida pela *Source*, sendo que quanto mais perto do obstáculo maior será a força aplicada pela normal.

Numa segunda instância a parte atrás do obstáculo onde na imediação não pode existir qualquer tipo de força do vento e conforme nos afastamos do obstáculo a força do vento irá recuperar cada vez mais a sua força original.

3.3 Resultados

Após a implementação do contorno de obstáculos criamos diversas cenas para ilustrar e visualizar o resultado final da nossa implementação.

Na figura 15 mostramos o comportamento do vento quando um obstáculo se encontra entre uma *Source* e um *Sink*. É de notar que as lâminas de relva estão a seguir paralelamente às faces devido à força de vento que lhes está a ser aplicada e não a um controlo de colisões das lâminas.

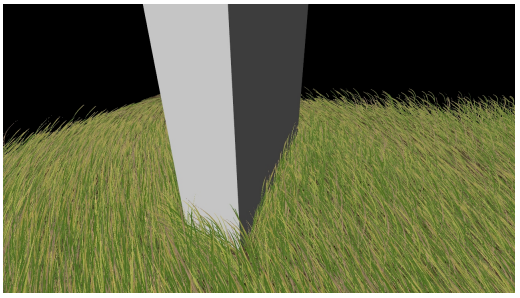


Fig. 15. Parede colocada entre um *Source* e um *Sink*

Na figura 16 observamos o comportamento de múltiplos obstáculos quando exposto a um vento uniforme constante. Nesta figura devemos notar a falta de vento na parte ocluída pelos obstáculos.

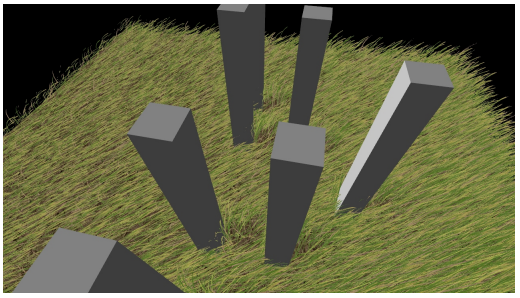


Fig. 16. Vários obstáculos opondo-se a um vento de direção constante.

REFERÊNCIAS

[1] Jahrmann, K., and Wimmer, M. Responsive real-time grass rendering for general 3d scenes. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2017), I3D '17, Association for Computing Machinery.

[2] Klasnja, P., Civan Hartzler, A., Unruh, K. T., and Pratt, W. Blowing in the wind: Unanchored patient information work during cancer care. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, Association for Computing Machinery, p. 193–202.

[3] Lo, Y., Chu, H.-K., Lee, R.-R., and Chang, C.-F. A simulation on grass swaying with dynamic wind force. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2016), I3D '16, Association for Computing Machinery, p. 181.

[4] Qiu, H., Chen, L., Chen, J., and Liu, Y. Dynamic simulation of grass field swaying in wind. *Journal of Software 7* (02 2012).