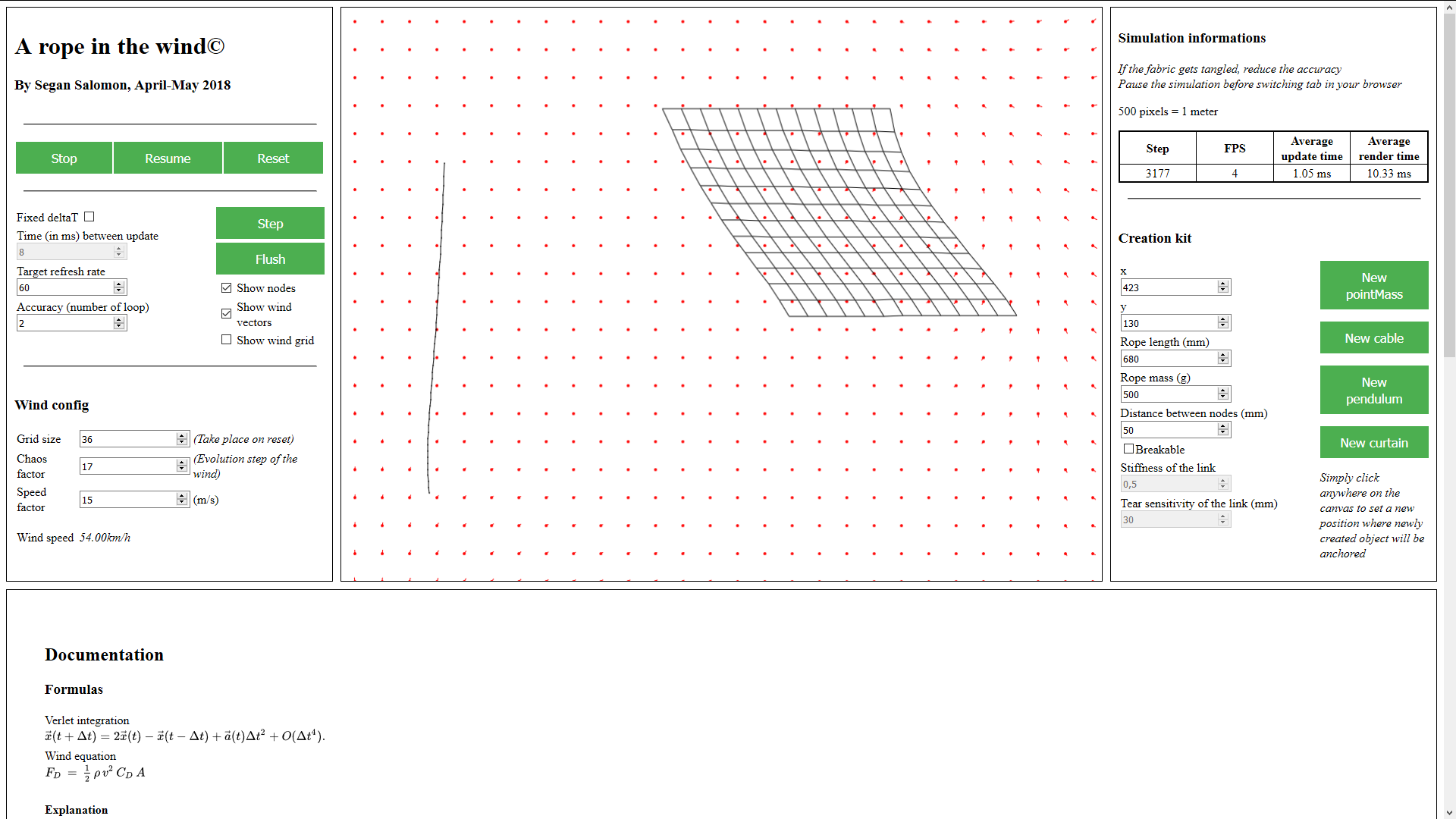
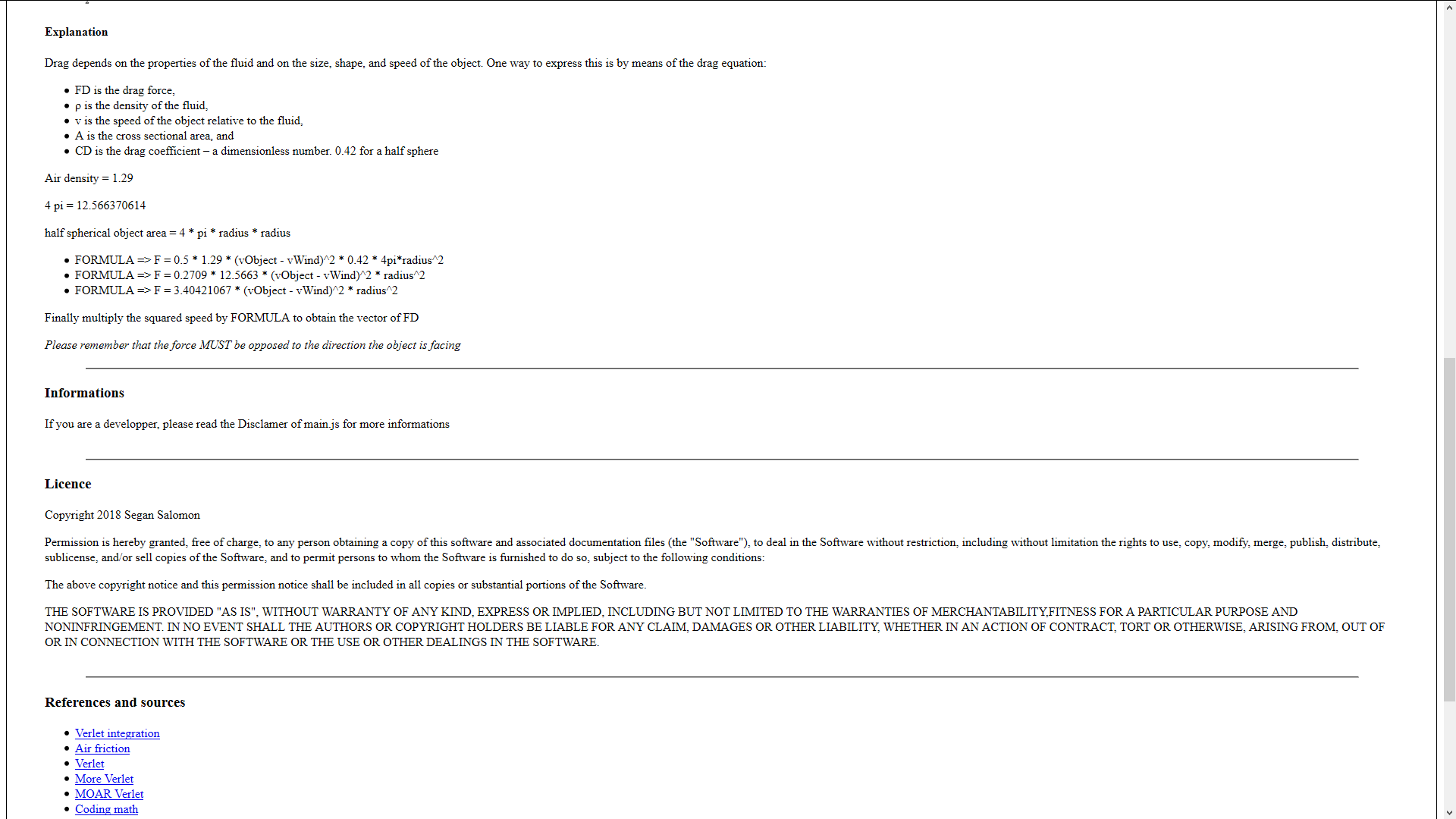
Une corde dans le vent

Algo. Num 2018 – mini projet individuel





Salomon Segan, le 12.06.2018

Nb d’heures passées au dév. : ~45h

## Résumé, pourquoi, comment et résultats

Ce projet vise à simuler le mouvement de mobiles de forme sphérique, reliés entre eux ou non, pouvant subir des forces extérieures et évoluant dans un monde physique simpliste.

La simulation d’une corde dans le vent est un sujet qui a retenu mon attention car je me questionnais : « Comment simuler un mouvement cohérent entre diverses particules reliées ? »

Le vent est généré au travers d’une bibliothèque fournissant un tableau 2 dimensions via le bruit de Perlin. Grâce à ces valeurs la génération de vecteurs de vent, représentant une vitesse en m/s est possible. Ces vecteurs sont ensuite répartis dans le monde où évolueront les masses.

Le vent évolue à chaque « tick » du programme, c-à-d à chaque appelle de la fonction nextStep recevant un paramètre un deltaT, cependant il est possible de choisir la fréquence de rafraichissement du vent au travers du paramètre « Chaos Factor » modifiant l’évolution des pas du bruit de Perlin.

Chacune des masses lors de sa recherche de prochaine position doit calculer sa vitesse relative dans l’air par rapport au vent où la vitesse du vent modifiable depuis l’interface graphique est soustraite à celle de l’objet.

Une fois la vitesse relative obtenue, elle est introduite dans la formule de frottement dans l’air afin d’obtenir un vecteur de force ajout à la masse. Grâce aux équations de Verlet sa prochaine position sera trouvée en utilisant uniquement son accélération directement déduit de sa masse et des forces subies et sa position précédente.

Lorsque la mise à jour de chaque particule est finie, la résolution des contraintes prend place. Cela consiste en la vérification du non dépassement des bordures du monde et l’inertie des masses reliées entre elles via un ressort plus ou moins élastique.

Les résultats obtenus sont visuellement convaincants et physiquement cohérents, ce qui m’a permis de fournir à l’utilisateur d’autres méthodes afin qu’il puisse créer un tissu, une corde, un élastique ou encore un pendule, tous obéissant les mêmes règles.

Deux illustrations permettant de valoriser votre travail :

1. Le modèle « théorique »
   1. Selon le modèle théorique et l’intégration par la méthode d’Euler, ici modifiée pour suivre la méthode de Verlet. Une position y1 d’un mobile dépendra de son accélération et de sa position précédente. Puisque le temps varie aléatoirement en fonction du temps il n’est pas possible de prévoir la position d’un mobile à un temps t donné.
   2. Les contraintes, c-à-d les ressorts entre mobiles répondent au travers du paradigme « Push-pull » ou chaque objet va tirer/pousser l’autre en fonction de sa propre masse et celle de l’autre.
2. Les résultats
   1. Les résultats obtenus sont purement visuels car utilisant l’approximation de la prochaine position via la méthode d’Euler, transformée par Verlet, cela serait bien trop volumineux à afficher pour chaque mobile évoluant dans le monde physique simpliste simulé.
3. Informations supplémentaires
   1. Concernant les heures de développement je tiens à préciser qu’il s’agit d’une estimation puisque je n’ai pas pris note du temps.
   2. Découpage du temps
      1. Conception ~8h
         1. Récolte d’informations globales ~5h
         2. Calculs des forces et formules ~3h
      2. Réalisation ~28h
         1. Interface graphique et canvas ~3h
         2. Développement du squelette ~5h
         3. Amélioration du squelette ~9h
         4. Ajout et validation du vent ~6h
         5. Ajout d’options supplémentaires ~5h
      3. Intégration ~9h
         1. Solidification du code ~3h
         2. Corrections d’erreurs ~4h
         3. Améliorations des performances ~2h