Dossier et Manuel

Simulation Bus – IN42

Nicolas Monneret
Alexandre Haffner
UTBM – GI02

Conception et mise en oeuvre

Eléments graphiques

Lignes et bus

Il a été convenu de créer les lignes à la main et non dynamiquement. Les lignes sont donc créées sur le clip mcCarte qui est la carte et non sur le _root. Ceci permet de profiter du zoom et du déplacement de la carte déjà implémentés.
Une ligne est composée d'un bus suivant une trajectoire définie manuellement. Des images clés sont situées à intervalles réguliers (définis dans les XML) sur cette trajectoire et symbolisent les arrêts de bus.
Une ligne de base invisible est donc créée sur la carte. Chaque bus de la simulation va être créé en dupliquant le clip ligne de base (duplicateMovieClip) et en faisant se déplacer le bus le long de sa trajectoire.

Passagers

Un passager est constitué de plusieurs frames qui correspondent au type de passager. Lors de la création du passager, on pourra ainsi spécifier son type et sa couleur en se plaçant sur la bonne frame (passager.gotoAndStop()). Tous les passagers de la simulation sont instanciés à partir de ce clip (attachMovie).

Timeline

La timeline est composée de 24 clips boutons qui correspondent aux 24 heures de la journée. Un code évènement est défini pour chaque bouton qui permet de modifier l'heure de la journée par click. Les boutons possèdent deux frames qui définissent deux états (2 couleurs). Ainsi on peut modifier la couleur du bouton si celui-ci correspond à l'heure de la simulation.

Date, Heure et échelle de temps

La date et l'heure sont définis dans deux variables globales. La date est une chaîne de caractères de la forme jj/mm/aaaa et l'heure est sauvée sous forme d'entier qui correspond au nombre de ms écoulées depuis minuit précédent. L'heure est affichée sous la forme hh:mm:ss via une fonction de formatage.
Il a été choisi de mettre le choix et l'affichage de la date sous forme d'une ComboBox. Celle-ci est remplie dynamiquement au chargement du fichier XML d'une ligne avec les différentes dates correspondantes aux arrêts des bus sur cette ligne.
Une ComboBox a également été mise en place afin de modifier l'échelle de temps et d'accélérer le défilement des heures. Le paramètre de vitesse est stocké dans une variable globale.
Des listeners permettent de modifier les variables globales date et echelleTemps en fonction des choix de l'utilisateur sur ces ComboBox.

Sélection des lignes de bus

Les anciennes checkBox du menu « Modes de simulation » ont été modifiées afin de faire apparaître les lignes de bus sélectionnées pour la simulation. Au click sur la case à cocher, si la ligne de bus n'a pas encore été chargée à partir du fichier XML, on la charge en instanciant une nouvelle ligne (new Ligne ()) et on remplit la liste de choix de date avec les dates de la ligne. Sinon on se contente de l'afficher ou de la masquer suivant l'état précédent.

Fenêtres d'informations

Des fenêtres d'information ont été créées afin de visualiser les informations des bus et des passagers en temps réel. Au survol d'un bus, on connaît ainsi le numéro de ligne, le sens et le nombre de passagers que contient le bus. Au survol d'un passager, on connaît le type de passager. On détecte le survol grâce à la fonction hitTest(). Lors du survol, on met la variable globale échelleTemps à zéro, ce qui permet de stopper le temps.

Clip Action

Un clip action a été créé afin de faire avancer le temps en temps réel à chaque frame grâce à la méthode getTime() de la classe Date qui retourne le timestamp UNIX. Les positions des bus des lignes visibles sont modifiés à l'aide de la méthode rafraîchir de la classe Ligne.

Simulation

Diagramme de classes

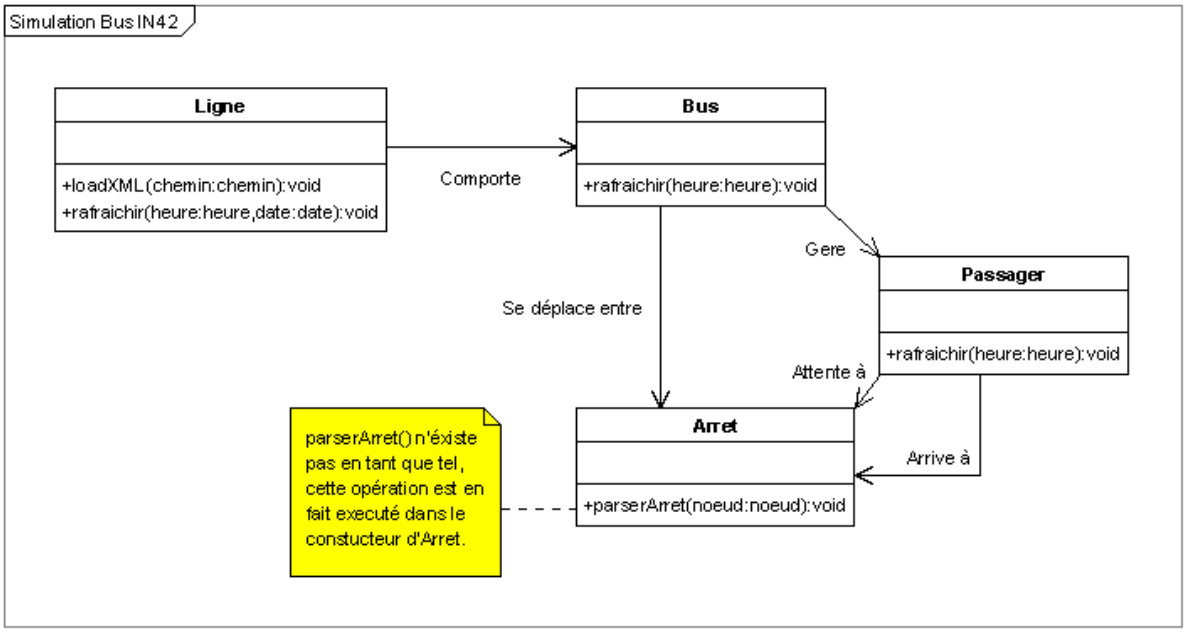
La partie simulation, entendez par là la gestion des positions et de l'affichage des bus et passagers en fonction du temps est géré par quatre classes. La classe « lien » avec la partie rendue est la classe Ligne. Des lignes peuvent être construites (c'est l'interface qui gère cette opération), à chaque ligne correspond une base de donnée sous forme d'un fichier XML que nous détaillerons plus bas.

Il est alors possible de « rafraîchir » une ligne, c'est-à-dire de placer correctement l'ensemble de ce qu'elle contient (bus et passagers) au bon endroit étant passés en paramètre l'heure et la date actuelle de la simulation.

La fonction « rafraîchir » de la ligne se charge d'appeler cette même fonction sur l'ensemble des bus, eux même appelant la fonction « rafraîchir » des passagers. Notez que le paramètre date n'est pas transmis au bus pour la raison suivante : les bus sont organisés dans Ligne grâce à un tableau associatif dont les clés sont les dates, il y a donc un bus n°1 pour chaque date par exemple. Autrement dit, une instance de bus ne peut se déplacer sur deux jours différents dans notre modèle (malgré que ce ne soit pas en réalité le cas, mais ça n'influe en rien sur la simulation : plusieurs instances de bus peuvent avoir le même numéro de bus).

Lorsque la ligne est créée le XML qui lui correspond est alors parsé, et ses informations recueillies. Lorsque l'on arrive sur un nœud de type « arret » il suffit de créer une instance

d'Arret qui va contenir l'ensemble des personnes qui vont être à un moment donné en attente à cet arrêt, de quoi connaître leur point de départ. Leur point d'arrivée est généré aléatoirement dans les arrêts suivant leur arrêt de départ après que le XML aie été complètement parsé.



Mouvements des bus

Comme nous l'avons vu c'est la fonction « rafraîchir » qui se charge de placer correctement les bus le long des guides. Nous avons utilisé l'interpolation pour arriver à nos fins.

Observez l'annexe 1.

L'annexe 1 représente le guide qui va nous servir à déplacer les bus. L'idée est de constituer un clip contenant le mouvement du bus le long de la ligne via un guide de mouvement de flash. Le clip du bus sera placé à chacun des arrêts et une interpolation entre ces arrêts sera alors réalisé. Il vous sera expliqué plus loin comment réaliser cette étape pour ajouter des lignes, mais le point important de cette étape est d'avoir exactement le même nombre de frames entre chaque arrêts (nb_frames).

L'intérêt est le suivant : pour placer le bus à l'arrêt « i » il suffit de se déplacer à la frame i*nb_frames + 1. Ce choix de conception a aussi été fait pour permettre une interpolation aisée entre les arrêts. Admettons que le bus soit à 6h à l'arrêt 1, et à 6h10 à l'arrêt 2. Nous savons quels sont les n° de frame de ces deux arrêts, nous savons aussi le nombre de frame qui les sépare. On peut donc à tout instant connaître la frame sur laquelle se situe le bus.

Prenons par exemple :
- nb_frames = 10 le nombre de frames entre chaque arrêts.
- On veut placer le bus à la bonne position à 6h02.

Il suffit de faire le rapport entre les deux heures et d'avance du nombre de frame correspondant.

- 6h10 – 6h00 = 10min.
- 6h02 – 6h00 = 2 min.
- 2/10 = 1/5ème
- 1/5ème * nb_frames = 2 frames.

Il nous faut donc avancer de deux frames depuis la frame de l'arrêt de 6h00. L'arrêt de 6h00 est l'arrêt 1, il est donc à la frame 11. Notre bus à 6h02 sera donc positionné en frame 13.

Cette interpolation a été aménagée de manière à permettre l'arrêt du bus aux arrêts auxquels quelqu'un attend. Le temps pendant lequel le bus s'arrête pour faire monter les passagers peut être modifié dans la classe « Bus » : variable static de classe TEMPS_ARRET.

Mouvements des passagers

Les passagers n'ont pas de mouvement à proprement parler. Ils apparaissent en deux endroits : à leur arrêt de départ, et à leur arrêt d'arrivée lors de la descente. Ces positions sont pré calculées lors de l'instanciation des « Passager » suivant l'heure à laquelle ils vont s'y trouver. A chaque passager est associé deux arrêts dont on connaît les horaires, on peut donc savoir où et quand ils seront affichés.

Les positions aux arrêts de départ et d'arrivée sont choisies aléatoirement dans deux carrés centrés sur les arrêts correspondants. Le rayon de ce carré peut être modifié dans les sources de la classe « Passager » : variable static de la classe « Passager » AIRE_PLACEMENT.
De même, il est possible de paramétrer l'avance avec laquelle le passager arrive à son arrêt de départ et le temps qu'il reste affiché à son arrêt d'arrivée après être descendu du bus : variables static TEMPS_AVANCE et TEMPS_ARRIVEE.

En passant l'heure lors du rafraîchissement des passagers il est alors possible de déterminer s'ils doivent être affichés à leur arrêt de départ, à leur arrêt d'arrivée, ou s'ils ne doivent pas être affichés.

Pour représenter les passagers dans le bus, nous avons fait le choix d'incrémenter un compteur propre à chaque bus indiquant s'il contient ou non des passagers, et quel nombre. Il est commun à tous les types de passagers mais pourrait être divisé en catégories. La fonction « rafraîchir » du passager retourne un booléen vrai si le passager est dans le bus qui vient d'en appeler le rafraîchissement. Il est alors possible pour le bus de contrôler le nombre de passagers qu'il contient actuellement.

Manuel utilisateur

Fonctions générales

Le démarrage du simulateur s'effectue via le menu « Affichage » puis « Modes » et en cochant la case « Mode Bus ».
Il est possible d'afficher la ligne de bus de son choix dans le sens de son choix en cochant la case correspondante du menu.
L'heure de la simulation peut être modifiée en cliquant sur la timeline ainsi que la date via la liste de choix.
Il est possible d'accélérer le temps via la liste de choix de vitesse.
Les informations sur les bus circulants sur le réseau et sur les passagers sont accessibles au survol de ceux-ci. On peut ainsi connaître le type de passager, la ligne de bus et le sens du bus.

Création de lignes

1) Créer un nouveau clip dans le dossier lignesBus de la bibliothèque nommé « ligneX_aller » ou « ligneX_retour » avec X le numéro de la ligne. Mettre sur un calque le clip mcCarte du dossier carte afin d'avoir un repère pour tracer le guide.

2) Créer un calque guide et dessiner le trajet des bus sur la carte.

3) Placer sur ce guide une occurrence du clip « bus » situé dans le même dossier. Nommer cette occurrence « bus ». Assurez vous de n'instancier qu'une occurrence de « bus » pour l'ensemble de la ligne. Dans l'éventualité où les passagers seraient tous toujours placés au début de leur ligne, c'est cette étape qu'il vous faudra recommencer.

4) Placer sur la première frame du guide le code suivant (à adapter suivant le cas) :

```
stop();
this.bus.numLigne = "Ligne X";
this.bus.sensLigne = "Aller ou Retour";
```

5) Créer les images clés correspondantes aux arrêts sur l'interpolation. Le nombre de frames entre deux images clés doit être constant, celui-ci sera indiqué dans le XML correspondant à la ligne.

6) Placer le bus aux différents arrêts aux frames correspondantes.

7) Supprimer le calque de fond qui représente la carte

8) Placer le clip ainsi créé contenant la trajectoire et le bus sur la véritable carte (mcCarte) située dans le dossier carte de la bibliothèque en superposant correctement la ligne à la carte. Donner à cette occurrence le nom que vous allez utiliser dans le XML pour la représenter et rendez-la invisible en y ajoutant le code suivant :

```
onClipEvent(load){
      this._visible=false;
}
```

9) Si la case à cocher du menu « Modes de simulation » n'est pas créée pour cette nouvelle ligne, créer la case « checkTjLigneX_aller » ou « checkTjLigneX_retour » avec X le numéro de la ligne. Placer sur la case à cocher le code :

```
on (click){
      // 0 : aller      1 : retour
      _root.affLigne(X,(0 ou 1), "chemin du XML");
}
```

Ajout de nouveaux code utilisateurs

L'ajout de nouveaux types d'utilisateurs se fait sur le clip « passager » situé dans le dossier « passagers » de la bibliothèque.

1) Créer sur le calque du clip « passager » une nouvelle image clé à la frame correspondant au code utilisateur (exemple : pour un code utilisateur 34, placer une image clé sur la frame 34).

2) Donner la couleur voulue au passager sur cette frame.

3) Ouvrir le clip « passagerPopup » situé dans le même dossier et ajouter dans le tableau « tab » de l'événement onClipEvent(load) la valeur « description de l'utilisateur » à l'indice correspondant au code utilisateur. (exemple : pour un code utilisateur 34, tab[34] = '' description '').

Base de donnée XML

Les données de la simulation sont contenues dans un ensemble de fichiers XML. Les données sont divisées par lignes, chaque ligne correspondant à un XML distinct. Notez qu'une ligne aller et une ligne retour sont considérées comme différentes car elles ne comportent pas forcement les même arrêts.

Le nom du fichier XML peut être modifié mais notre choix a été porté sur la forme suivante : « data_lignei_aller/retour.xml ». La ligne 1 aller aura par exemple pour fichier XML le fichier intitulé : « data_ligne1_aller.xml ».

Les fichiers XML de données pour la simulation n'ont pas besoin d'être correctement ordonnés mais doivent contenir l'ensemble des informations nécessaires, sans quoi il résultera un comportement indéterminé.

Voici la liste des règles concernant le XML :

```
<!ELEMENT ligne (numero_ligne?, nom_guideline, nombre_frames, arrets+)>
<!ELEMENT numero_ligne (#PC_DATA)>
<!ELEMENT nom_guideline (#PC_DATA)>
<!ELEMENT nombre_frames (#PC_DATA)>
<!ELEMENT arrets (arret, arret+)>
<!ATTLIST arrets date #REQUIRED>
<!ELEMENT arret (nom_arret?, indice_arret, numero_bus, heure, passagers*>
<!ELEMENT nom_arret (#PC_DATA)>
<!ELEMENT indice_arret (#PC_DATA)>
<!ELEMENT numero_bus (#PC_DATA)>
<!ELEMENT heure (#PC_DATA)>
<!ELEMENT passagers (#PC_DATA)>
<!ATTLIST passagers type #REQUIRED nombre #REQUIRED>
```

Dans ce XML est spécifié le nombre de frames qui a été utilisé pour séparer deux arrêts dans le guide correspondant, ainsi que le nom de ce guide dans le fichier flash.

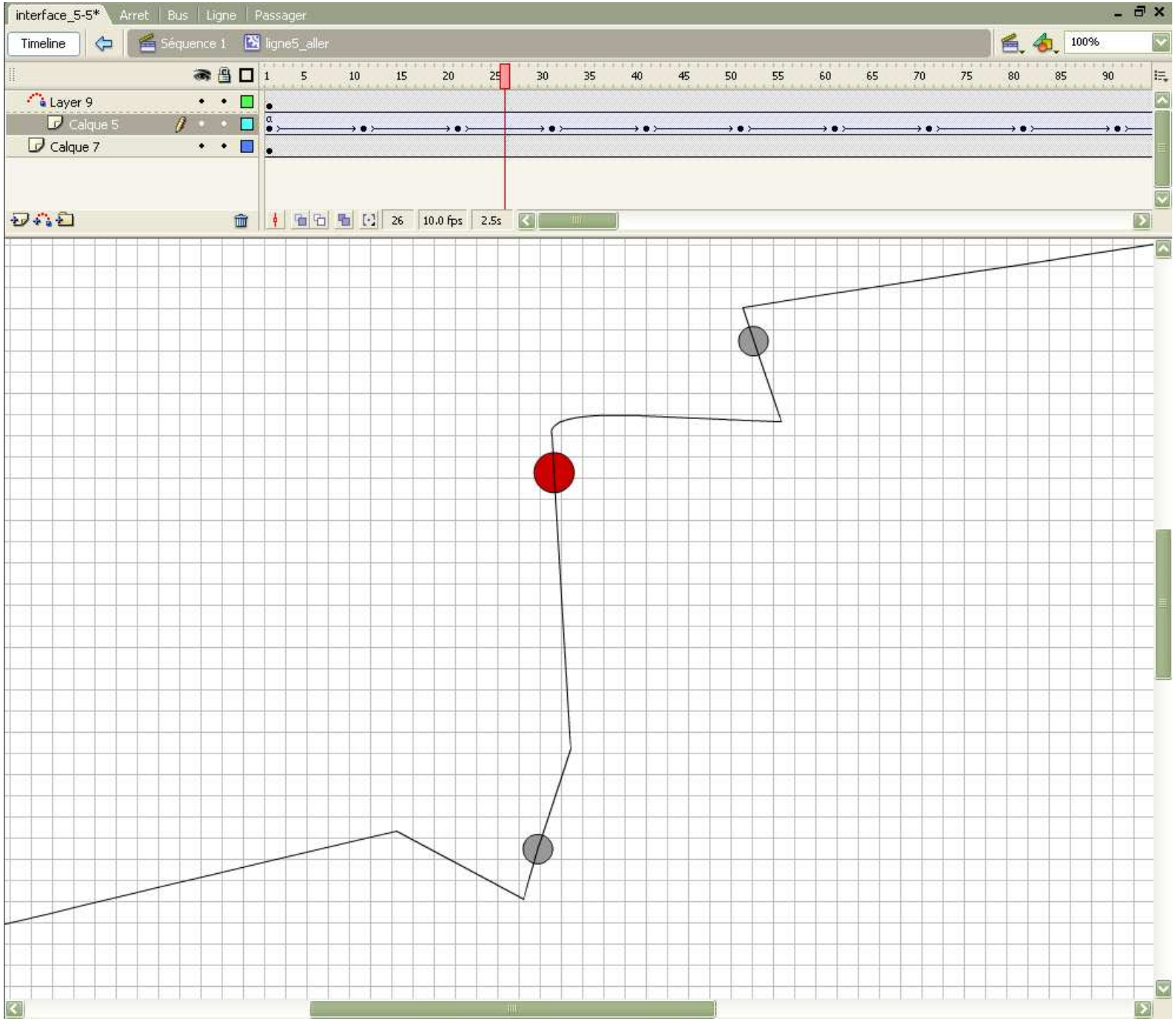
Une ligne contient un ensemble d'arrêts séparés par journées avec l'élément « arrets » qui regroupe les « arret ». Le nom de l'arrêt est facultatif, en revanche son indice et le numéro du bus concerné sont fondamentaux. Notez aussi que les heures doivent être cohérentes pour que la simulation fonctionne.
A chaque « arret » peuvent monter plusieurs passagers, ceux-ci sont séparés par types pour coller aux données initialement fournies (base de données Access).

Lors de la création de ce XML nous avons tenté de rendre aisé le passage depuis la base de donnée au XML. Certaine informations fondamentales n'étaient en revanche pas présentes dans les données initiales : le numéro du bus, et l'indice de l'arrêt. Autre point important, si vous voulez simuler correctement et ne pas voir disparaître le bus au dernier arrêt où attend un passager il vous faudra impérativement ajouter les horaires des arrêt de départ et d'arrivée de la ligne, même si personne ne monte à ceux-ci, ceci dans le but de permettre l'interpolation.

Vous trouverez en annexe 2 un exemple de XML que nous avons utilisé pour nos essais.

Conclusion

Plusieurs améliorations pourront être apportées à ce projet, surtout au niveau des données afin de pouvoir réaliser une simulation complète basée sur des données réelles.
Certaine parties optionnelles du XML pour le moment ne sont pas utilisées mais ont étés recueillies dans l'éventualité ou une future amélioration les utiliserait.


interface_5-5*
Arret
Bus
Ligne
Passager
Timeline
Séquence 1
ligne5_aller
100%
Layer 9
Calque 5
Calque 7
1
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
26
10.0 fps
2.5s