

Introduction à l'architecture d'un moteur de jeu vidéo

TP 1: Organisation initiale et instantiation de WebGL

Abdelkader Gouaïch

2023/2024

Etape 1: Instantiation du context WebGL dans une page HTML

Notre objectif est d'instancier un objet qui permet d'accéder aux fonctionnalités de WebGL. Cet objet est appelé un contexte WebGL et doit être récupéré et associé à un canvas HTML de notre page avant de pouvoir appeler les primitives de dessin graphique.

Description des étapes

- Récupérer le répertoire `session1/step1`
- Consulter le fichier `index.html`
- Ajouter un élément `html` de type `canvas` avec comme identifiant `webglCanvas`
- Ajouter un un élément `html` de type `script` pour pouvoir ajouter du code javascript directement dans page html
- Définir une fonction anonyme comme callback de l'événement `onload` sur l'objet `window`
- Code de la fonction anonyme:
 - Dans la fonction, déclarer une variable `canvas` initialisée avec un élément HTML dont l'ID est `webglCanvas` en utilisant la méthode `document.getElementById`.
 - Déclarer une variable `gl` sans initialisation.
 - Tentative d'Acquisition du Contexte WebGL:

- * Un bloc `try` est utilisé pour tenter d'acquérir le contexte WebGL de l'élément `canvas` et l'affecter à la variable `gl` en utilisant la méthode `canvas.getContext` et `webgl` comme identifiant de contexte
- * Si la méthode `canvas.getContext('webgl')` échoue, alors `canvas.getContext('experimental-webgl')` est appelée comme tentative alternative. La valeur de retour devra être affectée à la variable `gl`.
- * Si une exception est levée lors de ces appels, elle est capturée et ignorée grâce au bloc `catch` vide
- Commencer à travailler avec WebGL:
 - * Si la variable `gl` a été correctement initialisée (i.e., elle n'est pas `null` ou `undefined`) alors afficher un message log pour indiquer un succès; sinon indiquer une erreur.

Etape 2: Organisation du code

Nous avons utilisé une balise `script` pour mettre du code Javascript dans notre page html. Le but de cette étape est de ranger le code javascript dans un fichier externe qui sera référencer depuis la page html. Nous allons également utiliser une première fonction du contexte graphique qui permet de colorer l'arrière plan dans canvas.

Description des étapes

- Récupérer le répertoire `session1/step2`
- Créer un répertoire `session1/step2/src`
- Créer un fichier script `session1/step2/src/engine.js`
- Déplacer le code javascript d'initialisation du contexte vers ce fichier.
- Référencer le fichier script depuis la page html avec `<script type="module" src="./src/engine.js"></script>`
- Dessiner l'arrière plan avec une couleur:
 - Pour définir une couleur pour effacer le canvas, vous pouvez utiliser la fonction `gl.clearColor(r, g, b, alpha);`
 - Pour effacer effectivement le canvas nous pouvons utiliser : `gl.clear(gl.COLOR_BUFFER_BIT);`
 - Modifier votre code pour définir et effacer le canvas si le contexte est correctement initialisé

Etape 3: Première scène graphique

Pour cette étape nous allons donner commencer à réaliser des dessins primitifs dans un contexte GL correctement initialisé.

Nous vous proposons un projet déjà initialisé qui utilise des concepts que nous allons développer dans la suite du cours comme:

- les shaders
- le vertex buffer
- les entités et les composants

Nous vous demandons d'ouvrir et de modifier uniquement le fichier `src/myGame.js` pour réaliser cette étape.

Description des étapes

- Ouvrez le fichier `src/myGame.js`
- Ce script offre une fonction `function createPoint(x,y)` déjà réalisée qui permet de dessiner un point sur le canvas au coordonnées (x, y) . Le système de coordonnées utilise des flottants et va de -1 à 1 pour les deux axes.
- La fonction `function sceneSetup()` va contenir les instructions pour dessiner une scène graphique sur le canvas.
- En utilisant ces fonctions dessiner une scène qui reproduit les lettres: IUT comme le montre l'image ci-dessous.
- Vous pouvez créer des fonctions intermédiaires pour par exemple dessiner un segment composé de points.

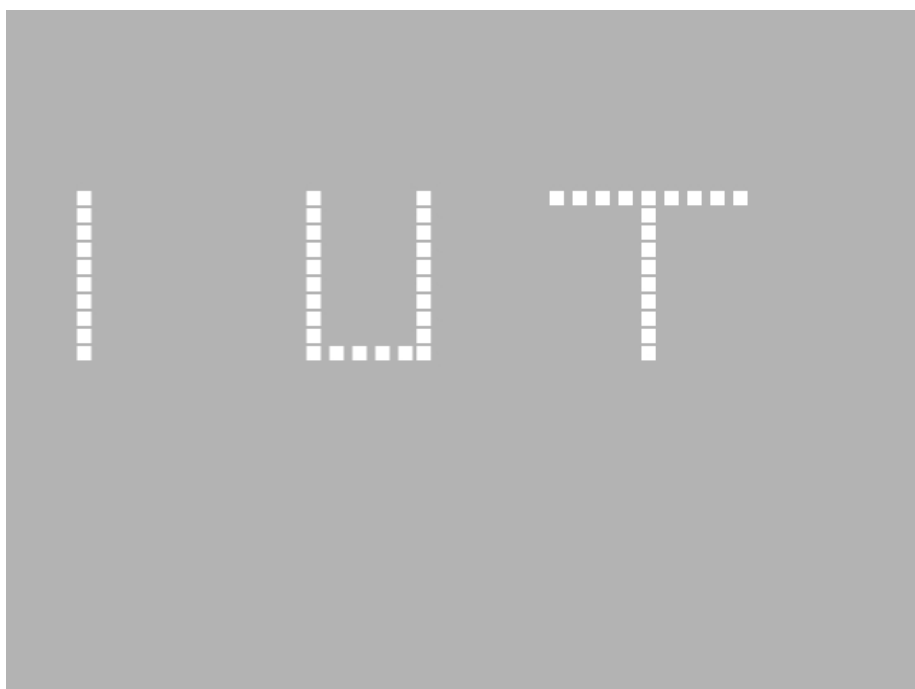


Figure 1: Scène à reproduire