

# Presentation projet Lua

Clement Lion – groupe Bogota

# Rappel critères d'évaluation du projet

Critères d'évaluation obligatoires
Découpage en modules : Le code est découpé en modules pour faciliter la compréhension, la réutilisation et la maintenance du code
Utilisation des machines à états : L'apprenant intègre une ou plusieurs machines à états pour gérer les comportements des éléments et/ou les états du jeu (Menus, gameplay, etc.)
Rotations et visées : L'apprenant maîtrise les fonctions trigonométriques de bases pour le déplacement et la rotation des éléments de gameplay (Sinus, cosinus, distance, calcul d'angles)
Utilisation des listes : Le jeu intègre l'utilisation des listes pour gérer les différents éléments du jeu (objets, ennemis, etc.) et pour faciliter la création de mécaniques de jeu complexes
Multiplés scènes : Le jeu comporte plusieurs scènes (Menu, gameplay, pause, etc.)
Fonctions : L'apprenant utilise des fonctions pour structurer et découper son code, et utilise les paramètres (valeurs, références).

# Jeu proposé

- SuperTank, un jeu dans lequel le joueur incarne un tank et doit détruire tous les tanks adverses afin d'aller au niveau suivant. Le jeu dispose d'un menu de départ et de trois niveaux. Détruire tous les tanks d'un niveau amène le joueur au niveau suivant. Si le joueur perd toutes ses vies ou termine tous les niveaux, il est ramené au menu de départ.
- Le joueur peut se contrôler au clavier ou à la manette (testé avec une manette XBOX One).
- Les tanks ennemis sont contrôlés par des machines à état (fonctionnement des états détaillé plus tard).

# Lancer le code du jeu

- Afin de lancer le code du jeu, il faut avoir installé le langage lua et le framework love sur votre ordinateur.
- Une fois dans la direction où vous avez copié le projet, exécutez la commande suivante : “love .”

# Commandes du jeu

- Commandes dans les menus :
  - clavier : Les flèches permettent de se déplacer et la barre d'espace de valider le choix.
  - manette : Les flèches et le joystick de gauche permettent de se déplacer et la gachette R1 de valider le choix.
- Commandes lors des phases de jeu :
  - clavier : La flèche "up" permet d'avancer, celle "down" de reculer, les flèches droites et gauches de tourner vers la droite et la gauche. La barre d'espace permet de tirer un missile et la touche "p" met le jeu en pause.
  - manette : La croix directionnelle gauche et le joystick gauche ont les mêmes effets : "up" permet d'avancer, "down" de reculer, droite et gauche de tourner vers la droite et la gauche. Le bouton R1 permet de tirer un missile et le bouton "start" met le jeu en pause.

# Structure de code

- assets : assets visuels du jeu.
- entities : entités du jeu. Désigne les composants d'un niveau (contrôlables ou non).
- graphics : objets pour gérer l'affichage des éléments à l'écran.
- levels : crée des objets "niveaux" et "menus" et les remplit d'entités.
- player : indique les éléments propres au joueur (contrôle, objet abstrait joueur).
- utils : fonctions utiles.

# Zoom sur l'entité "tank"

- Un tank est créé par un objet "tank\_factory". Il détient un "missile\_factory" qui lui permet de créer des missiles. Un tank possède un gestionnaire d'images qui lui permet d'être représenté à l'écran.
- Lors de sa création, le tank crée une liste de cercles qui le recouvre. Ces cercles forment sa hitbox.
- Le tank possède un état qui va conditionner la manière dont il va se comporter. Les états des tanks "ennemis" sont décrits dans les fichiers states>chase, states>idle, states>wait.
- Si le joueur contrôle un tank, alors le tank possède alors l'état "player".
- Les tanks sont chargés dans un gestionnaire d'entités (entity\_handler) qui se charge de les mettre à jour et de les dessiner.

# Diagramme d'état du tank

