Jeux sérieux

Technical design document
v1.0 (13/10/2020)

Kévin Alessandro, Iradukunda Valentin, Raynaud Jean-Baptiste
Lazlo Barragan, Romain Saclier

Dans le cadre de l'université de Montpellier

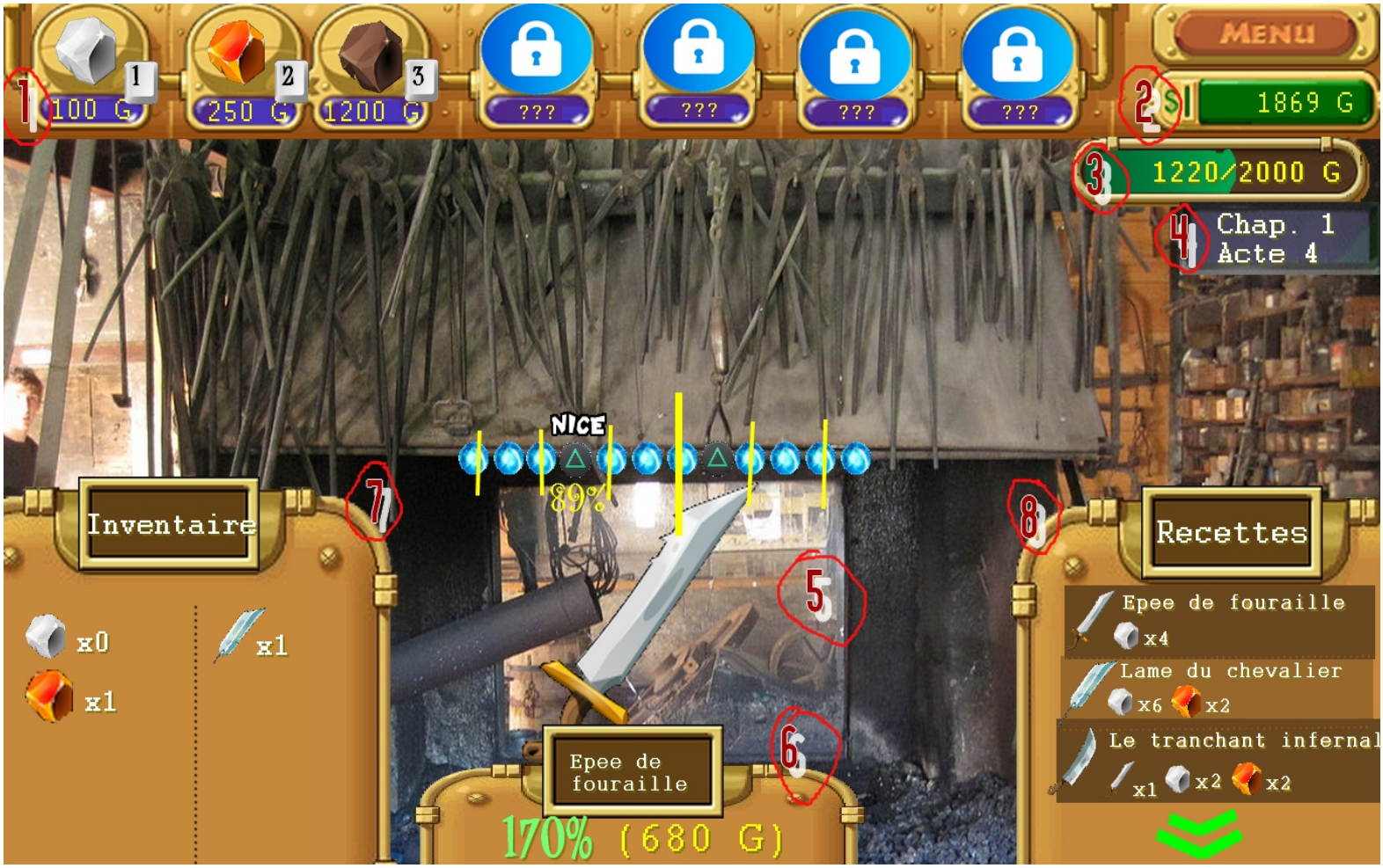
2020

Table of contents

0. Changelog

TDD v1.0

1. Storyboard et écrans



2. Diagramme Entity Component System (ECS)

Concernant la conception, nous prioritisons le coeur du gameplay, pour reculer par la suite vers la scène où ce gameplay évolue, pour reculer par la suite vers la racine du programme du jeu, qui dispose les scènes (scène “écran titre”, scène “in-game”, scène “partie terminée”, etc).

Nous proposons un concept technique du jeu en utilisant le pattern entité composant système. Ceci dit, les *rounds* sur notre jeu reposent beaucoup sur l'utilisation d'interfaces IHM, en cliquant sur des boutons ou parcourant des menus (achat de minerai, vente d'armes, sélection d'une recette...), et donc *pour ce style de*

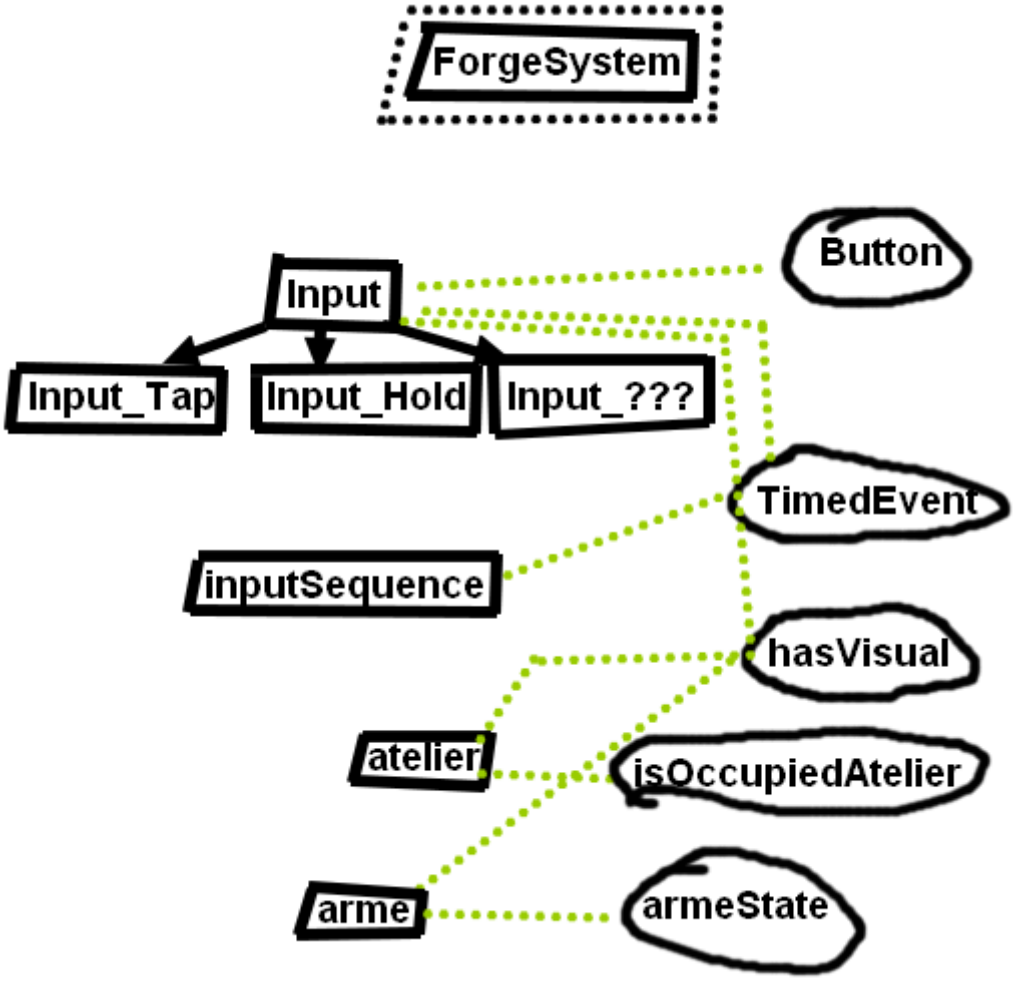
jeu, un design orienté autour des évènements (mouseClick, buttonPress...) pourrait être plus efficace. Ces évènements pourraient être aussi facilement utilisés pour la forge (les séquence de rythme, où le joueur doit appuyer avec un certain timing sur son clavier pour forger plus ou moins bien correctement une arme).

Une première version du ECS est ci-dessous.

On note avec un cadre ovale une composante, cubique une entité, et un cadre spécial le nom du système.

2.1. ForgeSystem

Gère le déroulement d'une forge : détecte les entrées et relâchements de bouton, actualise la qualité de l'arme en conséquence, fait défiler les entrées sur l'écran au cours du temps. Cela, pour chaque atelier (on peut avoir plusieurs forges en parallèles, une sur chaque atelier). Diagramme non exaustif / final.



Chaque *input* que le joueur doit presser utilise une composante *button,* qui permet de comparer l'état d'un bouton du clavier assigné à un input entre la frame t et t+1. Cela permet de détecter l'appui, le relâchement, ou la tenue d'une touche. Le jeu pourrait avoir des entrées de type *tap ou hold.* Les hold doivent être maintenus enfoncés pendant une certaine période. Potentiellement d'autres type d'entrées seront

possibles. L'input implémente timedEvent, cette composante a un attribut qui définit le temps avant que le timing pour un event soit parfait. ForgeSystem décrémente ce temps à chaque frame.

Chaque input est illustrée sur l'écran (la touche du clavier à enfoncer par exemple). hasVisual lui est retiré un certain temps après que le timing parfait est passé (“garbage collector”).

Toutes les input ne sont pas affichées en même temps, même si elles pourraient tenir sur l'écran. L'entité inputSequence contrôle le moment où une nouvelle image d'input doit être envoyée sur l'écran, c'est essentiellement un “Manager d'affichage d'input”. Elle utilise le temps grâce à timedEvent.

On peut forger sur chaque atelier en parallèle, un composante booléan isOccupiedAtelier permet de classifier les ateliers monopolisés de ceux libres pour la game loop.

Enfin une arme posée sur un atelier a un prix dynamique au cours d'une forge. Son sprite aussi potentiellement. L'arme doit aussi refroidir après une forge, ce qui monopolise un atelier.

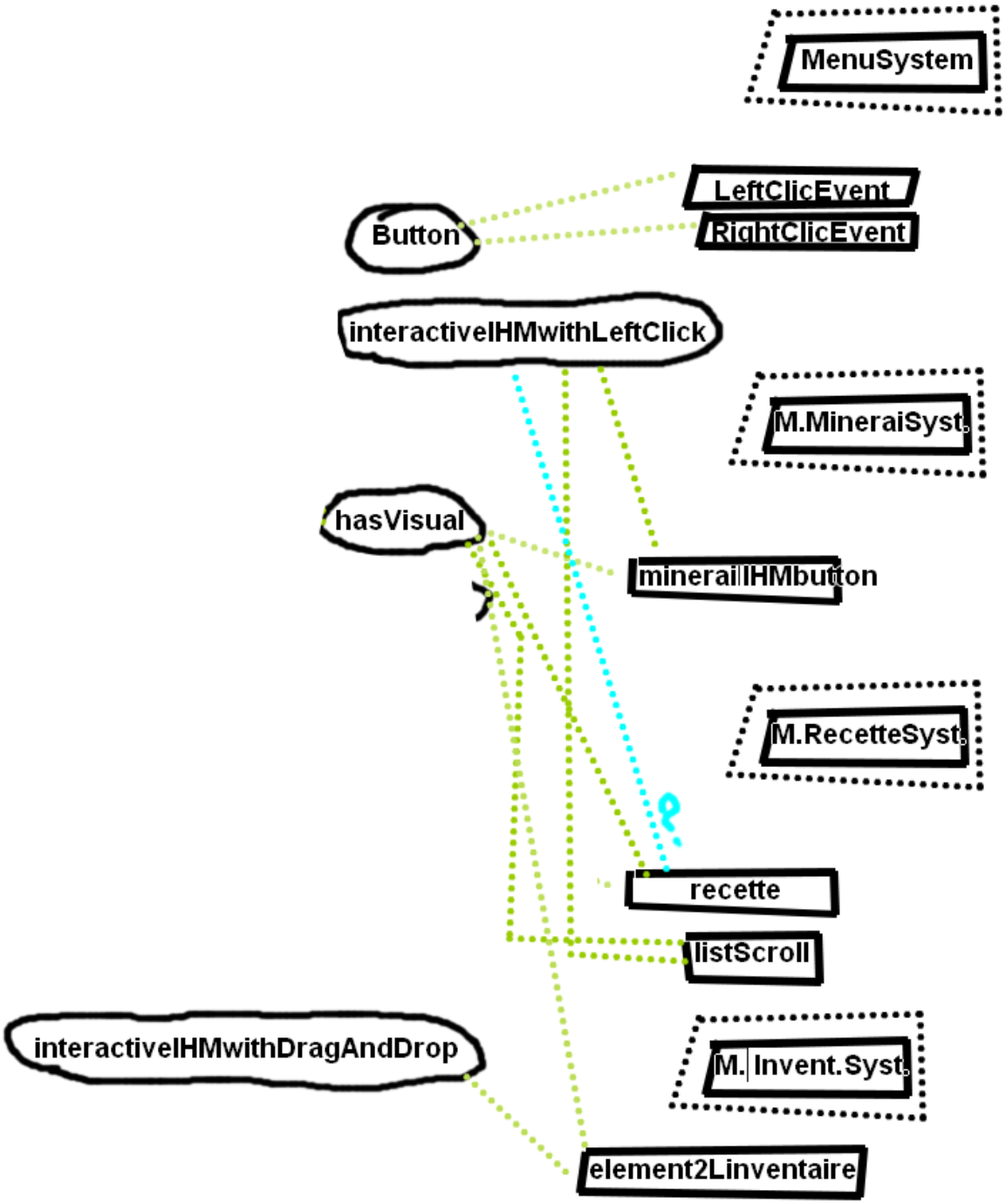
2.2. MenuSystem

Gère les évènements clic de souris et raccourcis claviers. Les principaux menus interactifs dans le jeu (in-game), sont : le menu des minerais (en haut), l'inventaire (en bas à gauche), et les recettes disponibles (en bas à droite).

Les raccourcis claviers ne sont jamais ambigüs, ils correspondent à une action précise. Le clic de souris cependant dépend de la région où le clic est effectué.

On se propose de détecter le menu ciblé par la souris, puis de transmettre l'évènement au sous-système correspondant : MenuMineraiSystem, MenuRecetteSysteme ou MenuInventaireSystème.

On considère l'évènement de clic de souris comme une entité. Elle utilise les mêmes méthodes de détection de pressage de boutons que les input de ForgeSystem.



Un tap de souris permet d'acheter un minerai. Un drag and drop permet de déplacer un élément de l'inventaire sur un atelier disponible. Peut-être que un tap de souris sur une recette permet de placer sur l'atelier automatiquement les ingrédients nécessaires à la recette. Une flèche permet de scroll dans la liste des recettes à forger quand il y en a trop à afficher à l'écran. On clique sur cette flèche pour scroll.

3. Spécification IHM

4. Technologie

Le jeu sera développé pour Windows uniquement, et non plus sur mobile, contrairement à ce que le GDD v1.1 mentionne. Le moteur et le langage de programmation restent à déterminer ; le plus logique dans ce cadre étant d'utiliser des outils que l'équipe de conception et développement connait. Selon le temps disponible réalistiquement le jeu sera réalisé sur du plus bas niveau (OpenGL/C++ par exemple) ou plus haut niveau (Unity par exemple).