AGE Engine - UML Diagram

Henry Jiang - CS246E

Registry
- entityManager: unique_ptr<EntityManager>
- componentManager: unique_ptr<ComponentManager>
- systemManager: unique_ptr<SystemManager>

+ createEntity(): EntityID
+ destroyEntity(entity: EntityID): void
+ addComponent<T>(entity: EntityID, component: T): void
+ emplaceComponent<T>(entity: EntityID, ...args: Args&&)
+ removeComponent<T>(entity: EntityID): void
+ hasComponent<T>(entity: EntityID): bool
+ getComponentType<T>(): ComponentType
+ registerComponent<T>(): void
+ registerSystem<T>(): T *
+ getRegisteredSystem<T>(): T*
+ setSystemArchetype<T>(archetype: Archetype&): void

EntityManager
- entityComponentMapping: unordered_map<EntityID, Archetype>
- entityCount: int

+ createEntity(): EntityID
+ destroyEntity(entity: EntityID): void
+ setArchetype(entity: EntityID, archetype: Archetype&): void
+ getArchetype(entity: EntityID): Archetype&

Entity
- entityId: EntityID
- registry

+ addComponent<C>(component: C)
+ emplaceComponent<C>(in ...args: Args&&)
+ hasComponent<C>(): bool
+ getComponent<C>(): C&
+ removeComponent<C>(): void
+ destroyEntity(): void
+ getEntityId(): EntityID

ComponentManager
- componentsArrays:
  unordered_map<ComponentID, unique_ptr<ComponentContainer>>
- componentTypes:
  unordered_map<ComponentTypes, ComponentID>

+ registerComponent<T>(): void
+ getComponentType<T>(): ComponentID
+ hasComponent<T>(entity: EntityID): bool
+ addComponent<T>(entity: EntityID, component: T)
+ emplaceComponent<T>(entity: EntityID, ...args: Args&&)
+ removeComponent<T>(entity: EntityID): void
+ entityDestroyed(entity: EntityID): void
- getComponentArray<T>():
  ConcreteComponentContainer<T>*

<<ComponentContainer>>

+ entityDestroyed()

ConcreteComponentContainer<T>
- components: list<T>
- entityComponentMapping: unordered_map<EntityID, list<T>:iter>
- componentEntityMapping: unordered_map<list<T>:iter, EntityID>

+ insertEntityComponent(entity: EntityID, component: T): void
+ emplaceEntityComponent(entity: EntityID, in ...args: Args&&)
+ removeEntityComponent(entity: EntityID): void
+ getComponent(entity: EntityID): T&
+ hasComponent(entity: EntityID): bool
+ entityDestoryed(entity: EntityID): void

SystemManager
- archetypes: unordered_map<SystemType, Archetype>
- systems: unordered_map<SystemType, unique_ptr<System>>

+ registerSystem<S>(in ...args: Args&&): S *
+ getRegisteredSystem<S>(): S *
+ setArchetype<S>(in archetype: Archetype&): void
+ entityDestroyed(entity: EntityID): void
+ entityArchetypeChanged(entity: EntityID, Archetype& type): void
- hasRequiredArchetype(test: Archetype&, parent: Archetype&): bool

System
entites: set<Entity>
registry: Registry *

+ getSystemArchetype(): Archetype

System

AsciiRenderSystem
- renderer: AsciiRenderer *

+ setRenderer(renderer: AsciiRenderer*): void
+ render(): void
+ getSystemArchetype()

<<ApplicationContext>>

+ init(): void
+ run(): void
+ stop(): void

CursesApplicationContext
windowWidth: int
windowHeight: int
manager: unique_ptr<CursesContextManager>
engineEventQueue: unique_ptr<EventQueue>
appEventQueue: unique_ptr<EventQueue>
renderTarget: unique_ptr<RenderTarget>
asciiRenderer: unique_ptr<AsciiRenderer>
sceneManager: unique_ptr<SceneManager>
eventlisteners: unordered_map<string, unique_ptr<EventDispatcher>>

+ init()
+ run()
+ stop()

Registry

SceneManager
- activeSceneName: string
- loadedScenes

+ createScene<S>(in ...args: Args&&): S*
+ unloadScene(name: string): void
+ getActiveScene(): optional<string>
+ setActiveScene(name: string): void

Scene
- sceneName: string
- registry

+ getSceneName(): string&
+ getRegistry(): Registry*
+ createEntity(): Entity
+ destroyEntity(in entity: Entity): void

+ init()
+ setup()
+ onActivate()
+ onDeactivate()
+ teardown()

AsciiRenderer
- width: int
- height: int
- screenBuffer
- renderTarget: RenderTarget *

+ drawCharacter(x, y, c): void
+ drawText(str): void
+ drawRect(x, y, w, h, fill): void
+ drawRect(x, y, w, h, fill, border
+ clear(): void
+ draw(): void

ScreenBuffer
- width: int
- height: int
- buffer: vector<string>

+ getWidth(): int
+ getHeight(): int
+ getBuffer(): vector<string>&
+ getBufferRow(row: int): string&
+ drawCharacter(x, y, c): void

<<RenderTarget>>

+ render(in buffer: ScreenBuffer &): void

CursesRenderAdapter
- cursesRenderer: CursesRenderer *

+ render()

CursesRenderer

CursesContextManager
- renderer
- keyboard

+ getKeyboardInstance()
+ getRendererInstance()

CursesRenderer
- width: int
- height: int
- window: WINDOW *

+ drawBuffer(in buffer: ScreenBuffer&): void
+ update(): void

CursesKeyboard
- keycodeBuffer: vector<int>

+ captureInputs(): void
+ pollKeycode(): int
+ flush(): void

EventQueue
- dispatchers: map<typeindex, vector<EventDispatcher*>>
- queue: vector<unique_ptr<Event>>
- eventType: vector<typeindex>

+ enqueue<E>(in ...args: Args&&): void
+ dispatchEvents(): void
+ registerEventDispatcher<E>(in dispatcher: EventDispatcher*
+ unregisterEventDispatcher<E>(in dispatcher: EventDispatch
+ unregisterAll<E>(): void

Event

CollisionEvent

BasicCollisionEvent
- entity1
- entity2

+ getCollidingPair() : pair

ApplicationEvent

KeyPressedEvent
- keycode: int

+ getKeyCode(): int

SwitchSceneEvent
- newSceneName

+ getNewSceneName()

EngineEvent

EngineDrawEvent

EngineInitEvent

EngineShutdownEvent
- code: int

+ getExitCode(): int

EngineUpdateEvent
- elaped: chrono::millisecond

+ getElapsedTime(): chrono::millisecond

<<EventDispatcher>>

*+ operator(in *event: Event, in *eventQueue: EventQueue)*
*- call(in *event: Event, in *eventQueue: EventQueue)*

FunctionEventDispatcher<E>
- function: std::function<void(E*, EventQueue*)>

+ call(): void

MemberEventDispatcher<T, E>
- instance: T
- func: (T::*func(E*, EventQueue*))

+ call(): void

AsciiRenderComponent
- prop: AsciiRenderProp*

+ getRenderProp(): AsciiRenderProp*

<<AsciiRenderProp>>

*+ render(out *renderer: AsciiRenderer, in xoffset: int, in yoffset: int): void*

CharacterProp
- character: char

+ render()

RectProp
- character: char
- width: int
- height: int

+ render()

TextProp
- text: string

+ render()
+ setText(in str: string)
+ getText(): string&

BitMapProp
- bitmap: BitMap

+ render()