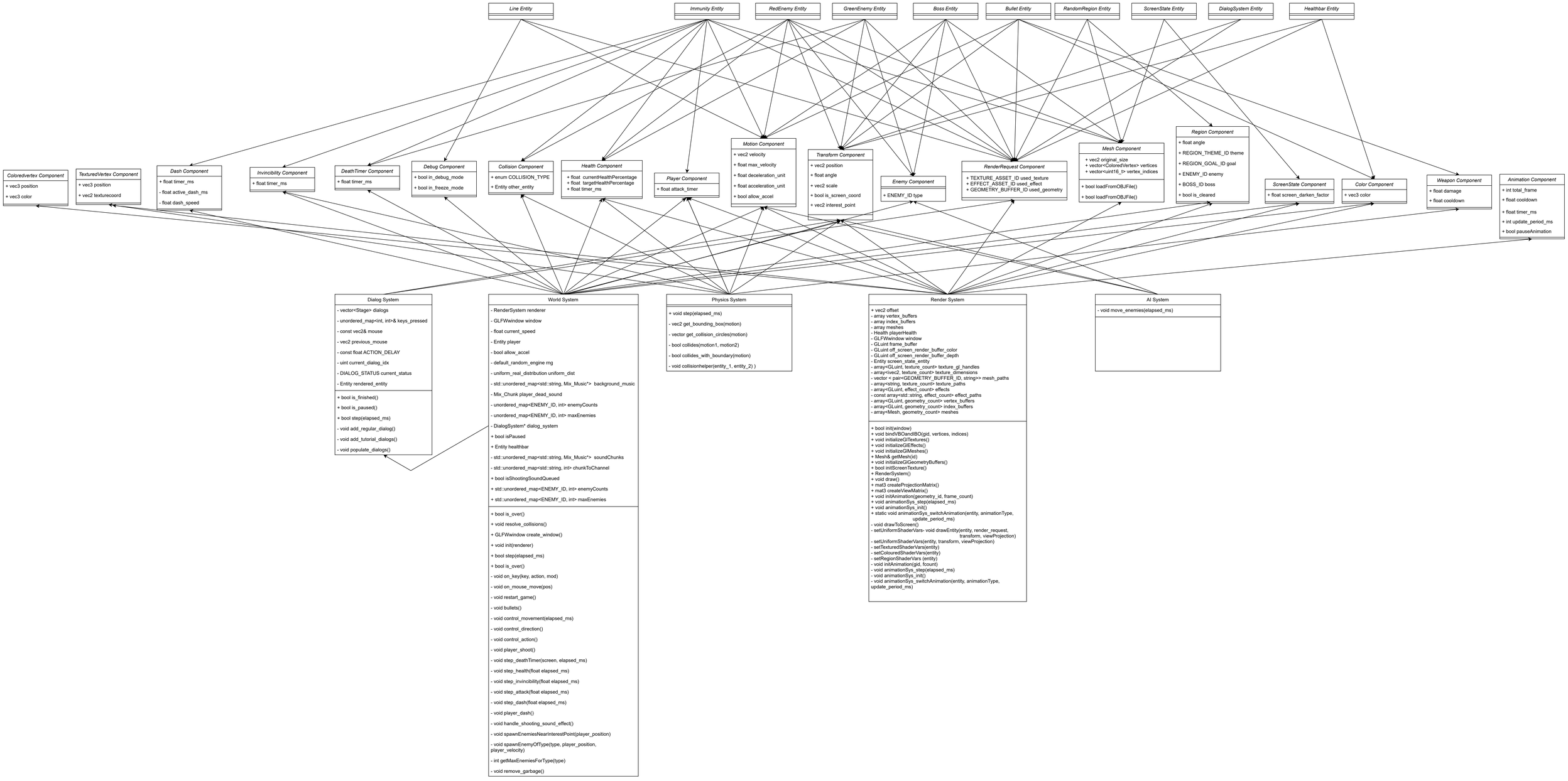
Line Entity
Immunity Entity
RedEnemy Entity
GreenEnemy Entity
Boss Entity
Bullet Entity
RandomRegion Entity
ScreenState Entity
DialogSystem Entity
Healthbar Entity
Coloredvertex Component
+ vec3 position
+ vec3 color
TexturedVertex Component
+ vec3 position
+ vec2 texturecoord
Dash Component
+ float timer_ms
- float active_dash_ms
- float dash_speed
Invincibility Component
+ float timer_ms
DeathTimer Component
+ float timer_ms
Debug Component
+ bool in_debug_mode
+ bool in_freeze_mode
Collision Component
+ enum COLLISION_TYPE
+ Entity other_entity
Health Component
+ float currentHealthPercentage
+ float targetHealthPercentage
+ float timer_ms
Player Component
+ float attack_timer
Motion Component
+ vec2 velocity
+ float max_velocity
+ float deceleration_unit
+ float acceleration_unit
+ bool allow_accel
Transform Component
+ vec2 position
+ float angle
+ vec2 scale
+ bool is_screen_coord
+ vec2 interest_point
Enemy Component
+ ENEMY_ID type
RenderRequest Component
+ TEXTURE_ASSET_ID used_texture
+ EFFECT_ASSET_ID used_effect
+ GEOMETRY_BUFFER_ID used_geometry
Mesh Component
+ vec2 original_size
+ vector<ColoredVertex> vertices
+ vector<uint16_t> vertex_indices
+ bool loadFromOBJFile()
+ bool loadFromOBJFile()
Region Component
+ float angle
+ REGION_THEME_ID theme
+ REGION_GOAL_ID goal
+ ENEMY_ID enemy
+ BOSS_ID boss
+ bool is_cleared
ScreenState Component
+ float screen_darken_factor
Color Component
+ vec3 color
Weapon Component
+ float damage
+ float cooldown
Animation Component
+ int total_frame
+ float cooldown
+ float timer_ms
+ int update_period_ms
+ bool pauseAnimation
Dialog System
- vector<Stage> dialogs
- unordered_map<int, int>& keys_pressed
- const vec2& mouse
- vec2 previous_mouse
- const float ACTION_DELAY
- uint current_dialog_idx
- DIALOG_STATUS current_status
- Entity rendered_entity
+ bool is_finished()
+ bool is_paused()
+ bool step(elapsed_ms)
- void add_regular_dialog()
- void add_tutorial_dialogs()
- void populate_dialogs()
World System
- RenderSystem renderer
- GLFWwindow window
- float current_speed
- Entity player
- bool allow_accel
- default_random_engine rng
- uniform_real_distribution uniform_dist
- std::unordered_map<std::string, Mix_Music*> background_music
- Mix_Chunk player_dead_sound
- unordered_map<ENEMY_ID, int> enemyCounts
- unordered_map<ENEMY_ID, int> maxEnemies
- DialogSystem* dialog_system
+ bool isPaused
+ Entity healthbar
- std::unordered_map<std::string, Mix_Music*> soundChunks
- std::unordered_map<std::string, int> chunkToChannel
+ bool isShootingSoundQueued
+ std::unordered_map<ENEMY_ID, int> enemyCounts
+ std::unordered_map<ENEMY_ID, int> maxEnemies
+ bool is_over()
+ void resolve_collisions()
+ GLFWwindow create_window()
+ void init(renderer)
+ bool step(elapsed_ms)
+ bool is_over()
- void on_key(key, action, mod)
- void on_mouse_move(pos)
- void restart_game()
- void bullets()
- void control_movement(elapsed_ms)
- void control_direction()
- void control_action()
- void player_shoot()
- void step_deathTimer(screen, elapsed_ms)
- void step_health(float elapsed_ms)
- void step_invincibility(float elapsed_ms)
- void step_attack(float elapsed_ms)
- void step_dash(float elapsed_ms)
- void player_dash()
- void handle_shooting_sound_effect()
- void spawnEnemiesNearInterestPoint(player_position)
- void spawnEnemyOfType(type, player_position, player_velocity)
- int getMaxEnemiesForType(type)
- void remove_garbage()
Physics System
+ void step(elapsed_ms)
- vec2 get_bounding_box(motion)
- vector get_collision_circles(motion)
- bool collides(motion1, motion2)
- bool collides_with_boundary(motion)
- void collisionhelper(entity_1, entity_2) )
Render System
+ vec2 offset
- array vertex_buffers
- array index_buffers
- array meshes
- Health playerHealth
- GLFWwindow window
- GLuint frame_buffer
- GLuint off_screen_render_buffer_color
- GLuint off_screen_render_buffer_depth
- Entity screen_state_entity
- array<GLuint, texture_count> texture_gl_handles
- array<ivec2, texture_count> texture_dimensions
- vector < pair<GEOMETRY_BUFFER_ID, string>> mesh_paths
- array<string, texture_count> texture_paths
- array<GLuint, effect_count> effects
- const array<std::string, effect_count> effect_paths
- array<GLuint, geometry_count> vertex_buffers
- array<GLuint, geometry_count> index_buffers
- array<Mesh, geometry_count> meshes
+ bool init(window)
+ void bindVBOandIBO(gid, vertices, indices)
+ void initializeGlTextures()
+ void initializeGlEffects()
+ void initializeGlMeshes()
+ Mesh& getMesh(id)
+ void initializeGlGeometryBuffers()
+ bool initScreenTexture()
+ RenderSystem()
+ void draw()
+ mat3 createProjectionMatrix()
+ mat3 createViewMatrix()
+ void initAnimation(geometry_id, frame_count)
+ void animationSys_step(elapsed_ms)
+ void animationSys_init()
+ static void animationSys_switchAnimation(entity, animationType, update_period_ms)
- void drawToScreen()
- setUniformShaderVars- void drawEntity(entity, render_request, transform, viewProjection)
- setUniformShaderVars(entity, transform, viewProjection)
- setTexturedShaderVars(entity)
- setColouredShaderVars(entity)
- setRegionShaderVars (entity)
- void initAnimation(gid, fcount)
- void animationSys_step(elapsed_ms)
- void animationSys_init()
- void animationSys_switchAnimation(entity, animationType, update_period_ms)
AI System
- void move_enemies(elapsed_ms)