SEPR 2019/20 Assessment 2

Team CheatCodez

Jonathan Grout, George Lesbirel, Cheuk Wang Wu, Lauren Quarshie, Muaz Atif, Lillian Coultas

Architecture

(V2 Updated From Assessment 1)

Any changes highlighted in green

Abstract Representation of Architecture

In this section we will look over our plan to the software structure of Kroy. We will cover an abstract construction of classes using Unified Modeling Language (UML), *Appendix 2.1*. Followed by a description and justification of the way we have designed it this way. Firstly, we will look over an abstract view of how our objects will interact with each other and the environment. In general we will use getters and setters for retrieving variables such as damage and hit points, although these have not all been included in the UML.

Our game will be played from a birds eye view. Players will be able to traverse York on a map spanning the entire region, which is not split into sections. There will be several homebases (Bases) for the player to protect, and alien bases (Landmarks) for the player to attack. Destroying Landmarks may trigger a minigame.

Our minigame consists of the user pumping water as fast as they can. Depending on: the difficulty they are on, length of playing and how much they pump, they will get different rewards and items. This minigame won't be triggered every time the player detroys a Landmark as we feel this could result in the user having too many bonuses at one time. The amount that the minigame is triggered with depend on how far through the game they are and difficulty. The user is given the option to skip the minigame. *Appendix 2.2* shows the behaviour of our minigame.

Figure 2.1 shows the UML diagram, a larger version can be found on the website. It is a basic visual representation of how the classes are related to one another and their purpose. This includes basic values that will be stored (such as hitpoints and position) and operations that each class should be able to perform. It shows a visual structure of the program and will be beneficial to speeding up the development of the game by providing an abstract document to refer back to.

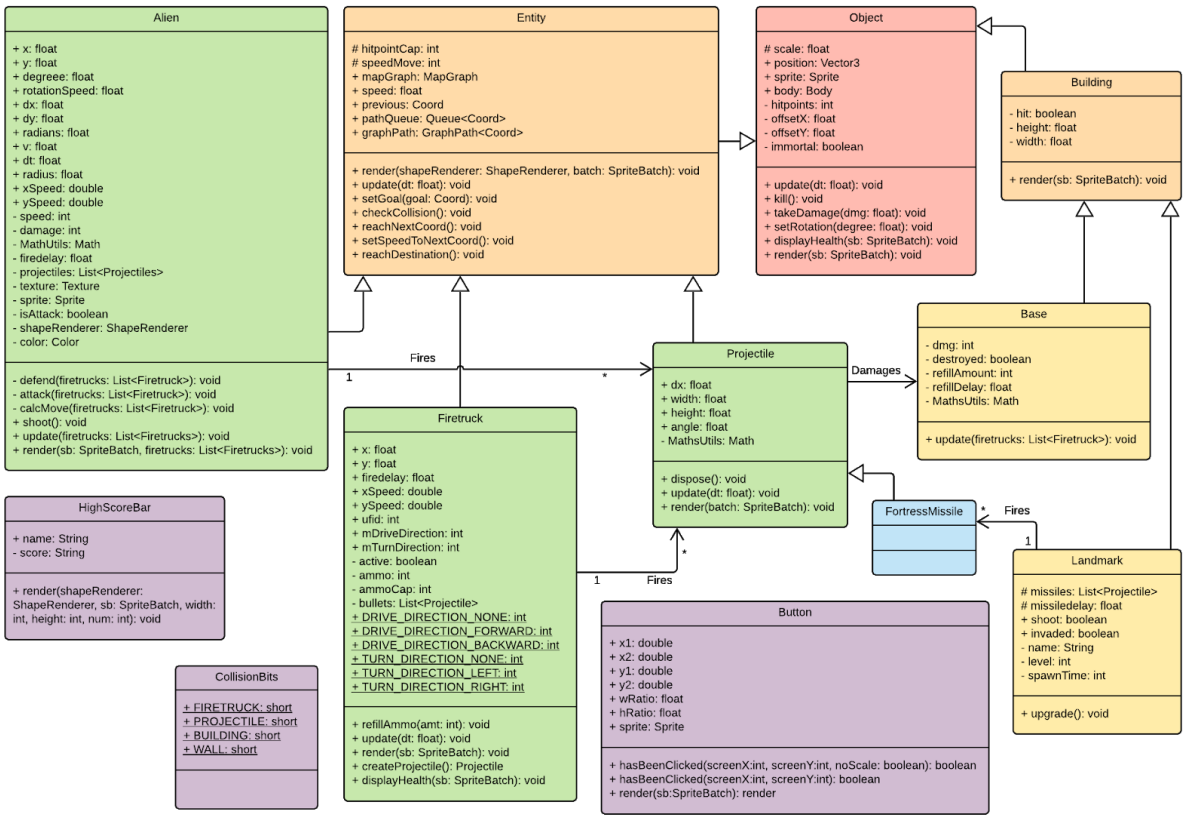

Figure 2.1 - The main game UML

Justification of Architecture

Software Component	Description/Justification	UML Explanation
Object	- Model is used to store the sprite models for each object - Position holds the position of the object in the game which will be mapped onto the libgdx coordinate system - The object class is the base class for most other classes in the game	No Composition relation with Stage, as Stage has been removed.
Entity	- Movement would store something like a transformation matrix for the movement of the entity - applied when move operation is next called - Spd_mov is the movement speed which would correspond to how many times the movement vector is applied - Hitpoints store the amount of health an entity has. - Move and take_dmg are functions that will be part of the interaction	

	with the game. The entity class inherits from the Object class. Entities are movable objects that can be interacted with e.g. Aliens can be attacked, they must be able to move and give damage.	
Firetruck	- Water volume for storing “ammo” - Water damage, how much 1 hit does - Speed - isActive - CPU_attack and CPU_defend are used to run the attack and defend operations on the fire trucks that aren’t controlled by the player - Inherits from Entity	CPU_attack and CPU_defend was used to show that attack and defend operations are also going to be implemented to be handled by the CPU Stage has been removed, so no association with it. Association with Projectile because Projectile is fired by Firetruck
UFO	Has been removed	Has been removed
Alien	- Master alien class - HP - Attack points - Inherits from Entity	Attack operation to show that aliens mainly attack enemies
Projectiles	- Damage dealt - maxDistance - (speed inherited) - Projectile is now a merging of Waterblast and Projectile	
Waterblast	Has been removed	Has been removed
FortressMissile	- FortressMissile is a renaming of Laserblast	
Building	- HP Basic class for structure objects that don’t move	
Base	Base for firetrucks to return to and refill water and repair damage	
Landmark	- Name Can be upgraded to increase health of Landmark, is a merging of Alienbase and Landmark	
Alienbase	Has been removed	Has been removed
Item	Has been removed, due to time/complexity	Has been removed

	constraints	
Stage	Has been removed	Has been removed
Map	Has been removed	Has been removed
HighScoreBar	- Scores names and scores of the high scores for the game - And methods to render them	No relations with any other class
CollisionBits	- Stores the collision bits for use in detecting collisions	No relations with any other class
Button	- Corner coordinates, height, width of the button - And methods to tell if it's been clicked or not - 2 methods using overloading to help with scaling	No relations with any other class