

ENG1 Team Project

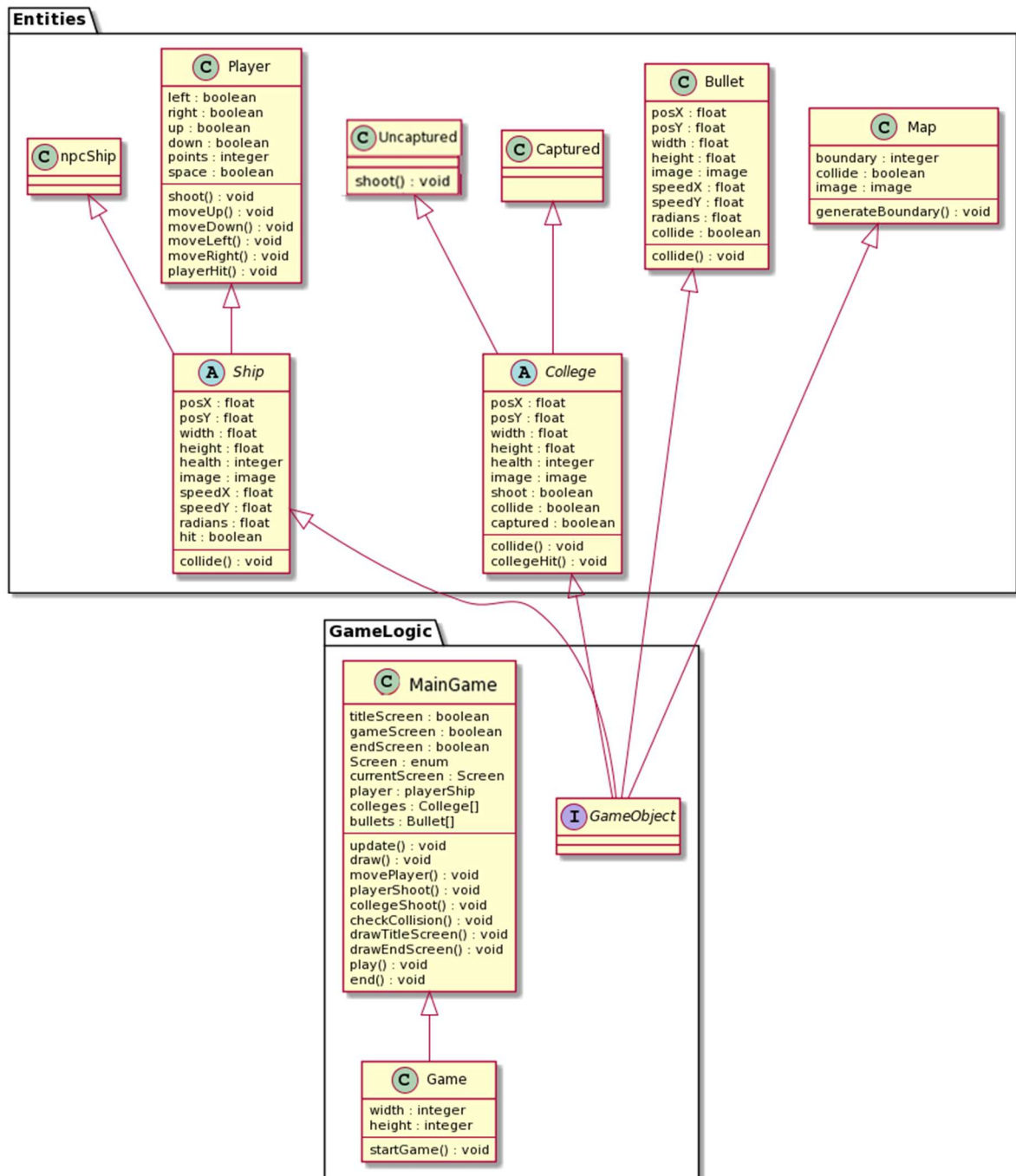
Deliverable 3: Architecture

Team 13 – Team Unlucky

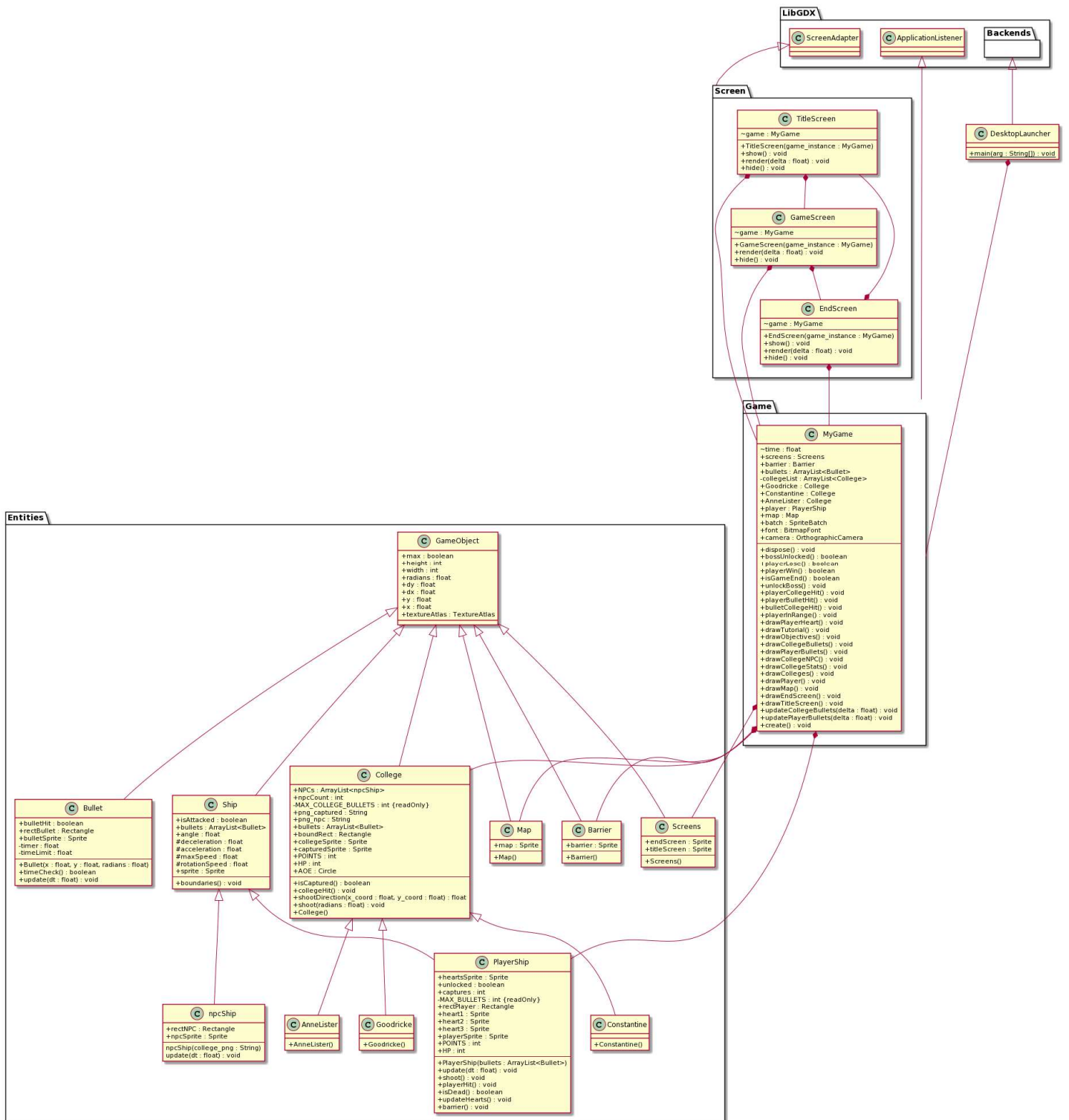
LILLY BROWN
ROSCOE GILLATT
ADAM JOHNSON
MATTHEW TAYLOR
BRANDON WEST
AYMAN ZAHIR

a) The tool we used to produce the abstract and concrete architecture was PlantUML, this allows a user to create diagrams and flowcharts from a plain text language, we chose to produce a class diagram.

Abstract Architecture:



Concrete Architecture:



b)

We used our abstract architecture to inform our implementation of code.

We designed our architecture using OOP-Entity Hierarchy.

At first we reviewed what actors would be within the game. Eg, the player includes enemy ships, the colleges, the bullets and the map. And the game logic needed to run such as displaying the sprites and switching between screens

College: is the class for each static college in the game with an uncaptured mode which will allow the college to engage the player with weapons and a captured command which

Ship: handles both enemy and player ships allowing movement around the map and firing

Bullet: functions as a projectile for the ships and colleges to fire.

Map: generates the boundary and collision for the in game arena.

As we were implementing our initial plans had to change due to working and optimising our libGDX code. Resulting in a much larger concrete architecture. Although there is a large disparity between the abstract and concrete architecture many of the

- change from closed architecture to semi-open architecture

- mention change from if else game screen change to libGDX's game + screen adapter implementation

- MyGame acts as facade + singleton (benefits)

- mention coupling + cohesion

- use of sprites and texture atlas under gameObject()

- using functions instead of flags → reduces overhead

Concrete architecture justification

<u>Architecture construct</u>	<u>Requirement ID</u>	<u>How it is fulfilled</u>
PlayerShip, College	UR_GAMEPLAY_MODE	MyGame initializes one PlayerShip and Colleges from a list of colleges. The update method in PlayerShip handles player movement and shooting.
MyGame, DesktopLauncher	UR_DESKTOP	The main function in DesktopLauncher initializes a new desktop application with a new instance of MyGame upon launching the game
MyGame, BitMapFont	UR_USAGE_MODE UR_EASE_GAMEPLAY. FR_TUTORIAL, NFR_USABILITY	MyGame's drawTutorial method triggers when new game is created and overlays timed BitMapFont based instructions
GameScreen, Orthographic Camera	UR_DISPLAY NFR_SCREEN_COMPA TIBILITY	GameScreen has code in render function which tracks the screen dimensions with the camera dimensions to preserve aspect ratio
PlayerShip	UR_INTERACTION	PlayerShip is controlled through keyboard input in the update function using Gdx.Input.Iskeypressed()
Map, MyGame	UR_MAP	Map initializes a new sprite that is designed

		to look like a lake and the sprite is then loaded by GameScreen's render function
PlayerShip	FR_ONE_SHIP	The player uses the keyboard to control only the player ship
PlayerShip, Bullet	FR_FIRE	PlayerShip's shoot() initialises an instance of the class Bullet and adds the bullet to the list of bullets in MyGame.
College, Circle, MyGame, PlayerShip	FR_BUILDINGS	When a College is initialized it has a Circle with a set radius. The method playerInRange under MyGame checks if PlayerShip overlaps with the Circle in which case college shoots at the PlayerShip
TitleScreen, Endscreen	FR_MENU	TitleScreen() and EndScreen() initialise short clear menu screens to start and reset the game
MyGame, Player, College	FR_ANIMATION	The method drawPlayerHearts draws the varying heart sprites of the player based on the player's current HP. The method drawColleges draws a collegeSprite or capturedSprite based on whether the college is captured or not.
Ship, PlayerShip, Barrier	FR_BOUNDARIES	Ship's boundary method and PlayerShip's barrier method define the bounds of the map when the boss is unlocked and locked respectively to restrict player movement.
College, PlayerShip, MyGame	FR_POINTS	PlayerShip increments POINTS by the College POINTS once a it's captured in the bulletCollegeHit() function under MyGame
GameScreen	FR_ONE_SCREEN	All the GameObjects initialized in MyGame is rendered only in the GameScreen
MyGame, PlayerShip, Barrier	FR_TASKS	MyGame's drawObjectives() function displays the task of capturing colleges, displays how many colleges are captured based on PlayerShips captures, and only displays the final objective when the task of capturing all other colleges is complete. The boss college is unreachable initially due to the barrier restriction on the player
GameObject, College, PlayerShip, npcShip	FR_COLORBLIND	All GameObject sprites images are custom made for each College, PlayerShip and npcShip with different images so that they are identifiable without color