3. Architecture [22 marks]

Group 11 - 11 Musketeers

Osama Azaz

Adam Dawtry

Tom Jackson

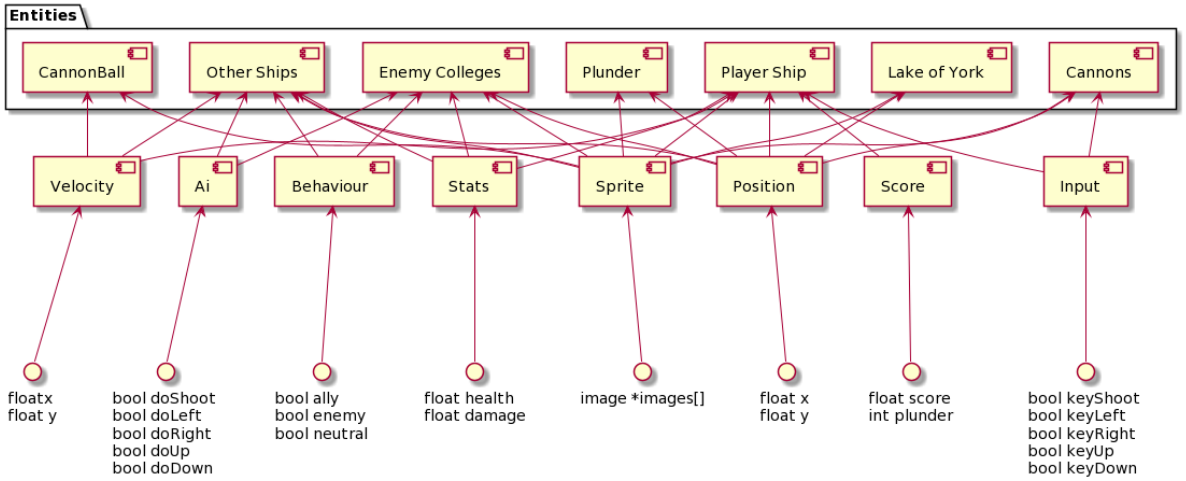
Brendan Liew

Holly Reed

Harry Ryan

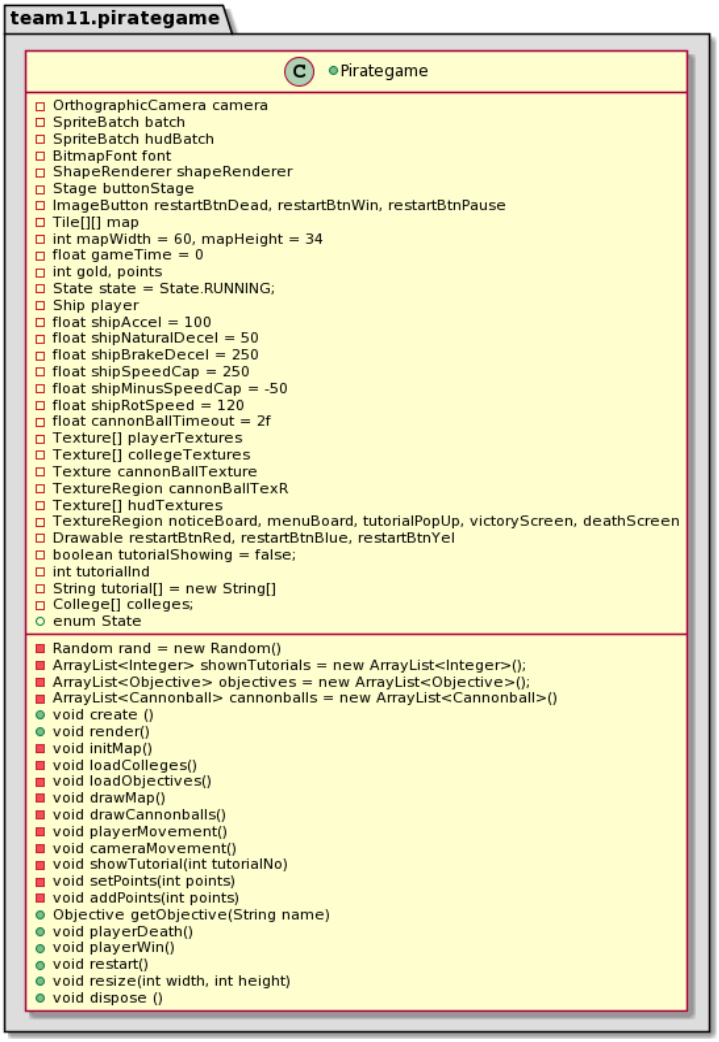
a)

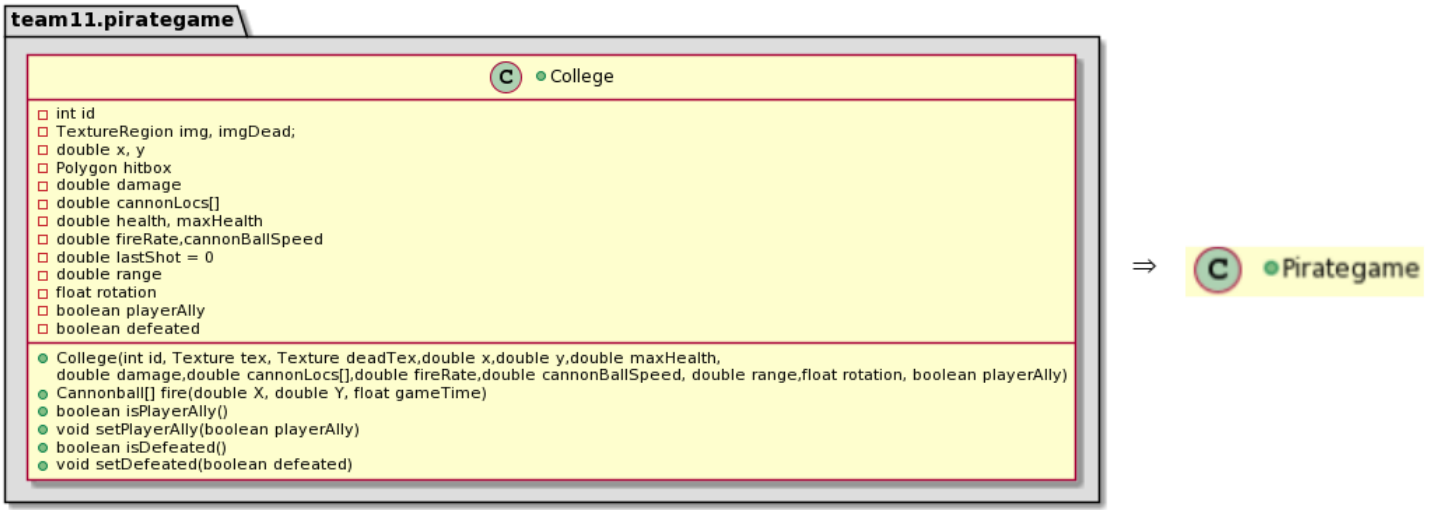
Abstract Architecture using Component Diagram:

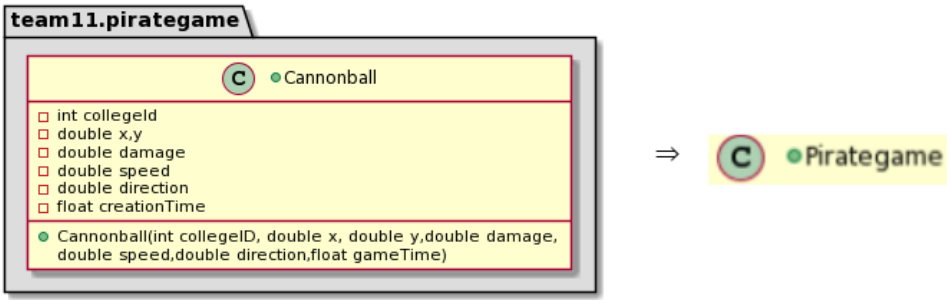


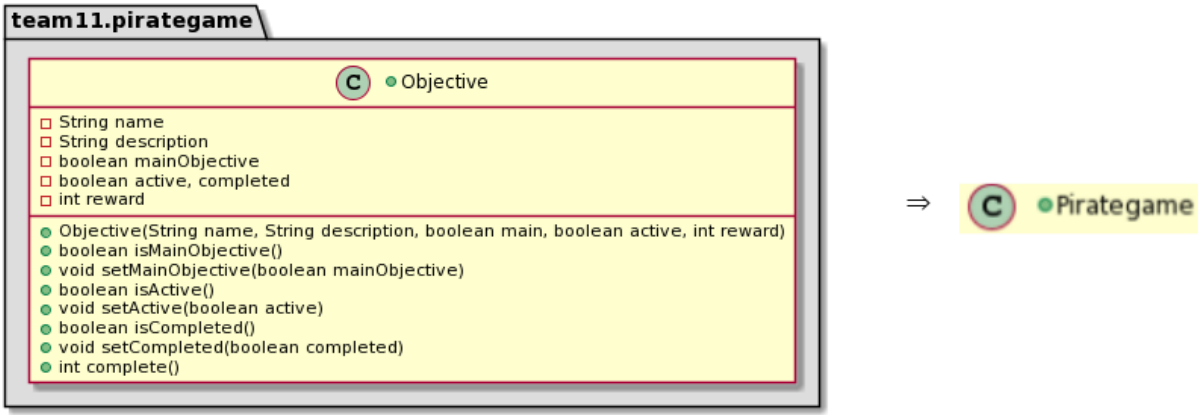
- Component diagram is used to relate each of the main components to relevant entities and display the game logic systematically, this is then built upon during the design process and developed into concrete architecture.

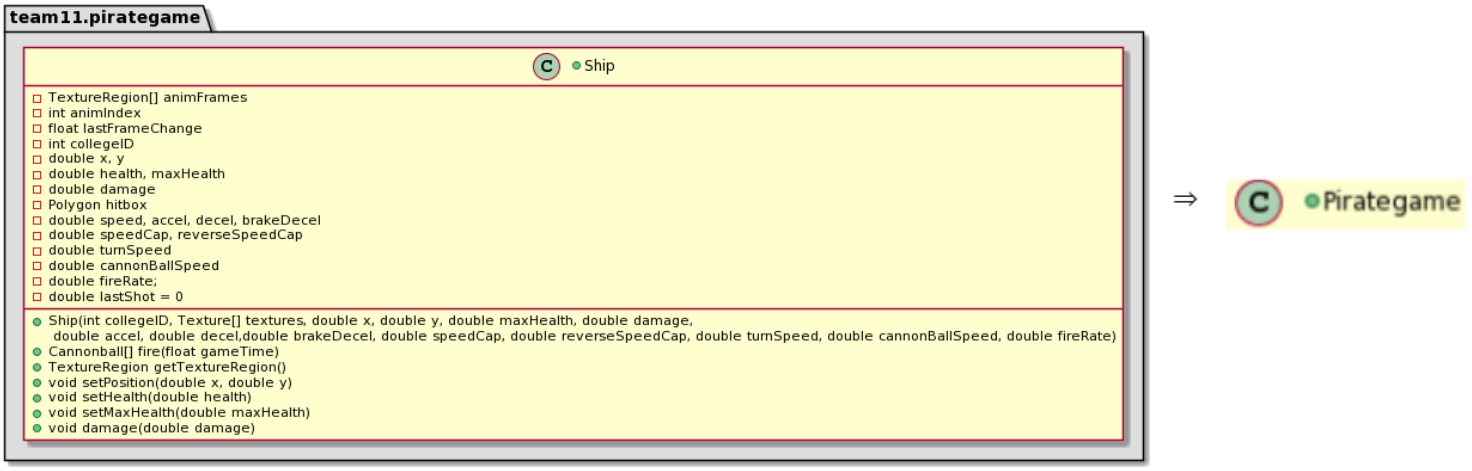
Concrete Architecture using Class Diagram:

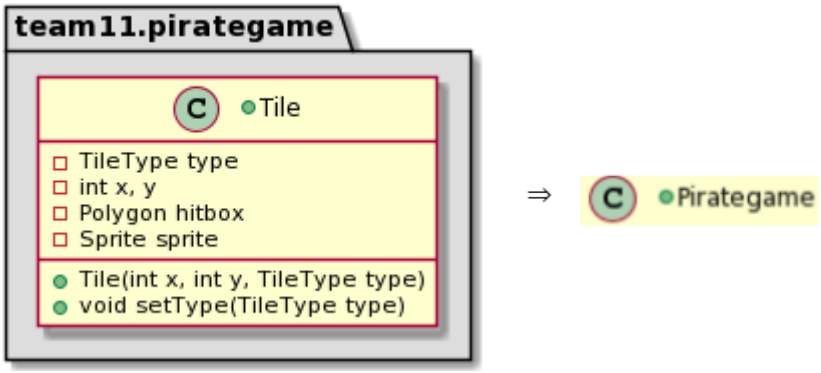


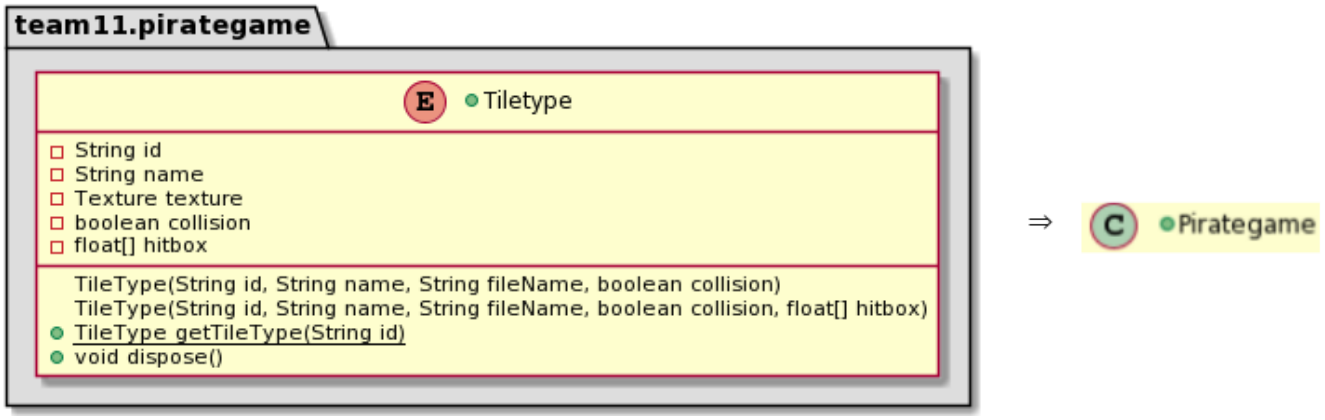

team11.piratego
College
int id
TextureRegion img, imgDead;
double x, y
Polygon hitbox
double damage
double cannonLocs[]
double health, maxHealth
double fireRate,cannonBallSpeed
double lastShot = 0
double range
float rotation
boolean playerAlly
boolean defeated
College(int id, Texture tex, Texture deadTex,double x,double y,double maxHealth,
double damage,double cannonLocs[],double fireRate,double cannonBallSpeed, double range,float rotation, boolean playerAlly)
Cannonball[] fire(double X, double Y, float gameTime)
boolean isPlayerAlly()
void setPlayerAlly(boolean playerAlly)
boolean isDefeated()
void setDefeated(boolean defeated)
Piratega


team11.piratego
Cannonball
int collegeId
double x,y
double damage
double speed
double direction
float creationTime
Cannonball(int collegeID, double x, double y,double damage,
double speed,double direction,float gameTime)
Piratega


team11.piratego
Objective
String name
String description
boolean mainObjective
boolean active, completed
int reward
Objective(String name, String description, boolean main, boolean active, int reward)
boolean isMainObjective()
void setMainObjective(boolean mainObjective)
boolean isActive()
void setActive(boolean active)
boolean isCompleted()
void setCompleted(boolean completed)
int complete()
Piratega







- A class diagram is used to describe the concrete architecture, it allows detailed modelling of the relationship between classes, their attributes and operations
- This forms the architecture when designing and constructing the source code of the game

b)

Requirements ID	Abstract Description	Concrete Description	Justification
FR_PLAYER_HEALTH	stats(float health)	void setHealth(double health), double getMaxHealth(), void setMaxHealth(double maxHealth)	The static “float health” is developed to “maxHealth” & “getMaxHealth()” to check if health has depleted to execute game loose sequence
UR_POINTS & FR_PROGRESSION	Float score	Int points Private void setPoints(int points) Private addPoints(int points)	In the user requirements , we are required to add a point system for user to power up the ship. From abstract, we need to have a function to store the points gained and also add points as the game progresses to allow ships to be upgraded
FR_GAMEPLAY_INSTRUCTIONS & UR_INTUITIVE	N/A	String tutorial[] = new String[] ArrayList<Integer> shownTutorials = new ArrayList<Integer>(); Private void showTutorial(int tutorialNo)	The requirement for user interaction with the project is user instructions to operate the game, this is only developed in the concrete architecture as concrete considers user requirements as well as gameplay elements such as entities. This is done by displaying a tutorial at when the game starts
UR_ONE_SHIP	Input(bool keyShoot bool keyLeft bool keyRight bool keyUp bool keyDown)	Class Ship Private void playerMovement()	Users can only use one ship to navigate through the map.Abstract had the input for the ships made.When making concrete, We implemented a class for ship which generates the ship and its other features such as movement speed and the function is to allow players to move the ship around the map
UR_SIDE_OBJECTIVES	N/A	Class Objective Public Objective getObjective(String name)	The requirements for the user in this game is to have other objectives besides the main objective. Abstract only focuses on main objective so this is developed in concrete architecture by making a class for objectives which includes the description,rewards and the availability of the objectives.The function also displayed the objectives on the screen to remind users of the objectives
UR_LOOSEABLE	Float health =0	Public void playerDeath() Public void render()	User must have a chance to lose this game,when the ship has ran out of health, the player loses.From

			abstract,we have extended it into a function to report a player death as a state where when the player dies, the game stops running and displays a death message
FR_REAL_TIME	N/A	private float gameTime = 0; Public void render()	The requirements is for the game to run in real time.From abstract,we have extended in concrete which we have a value 0 stored in a variable gameTime at start of game and the function render() is ensure the game records every single action taken by ships at gameTime
UR_RESTART,	N/A	Function restart()	The requirement for the game to be able to restart met when either the player loses or wins the game then a restart button is button is displayed, clicking on this calls the restart function which reinitialise everything
FR_STATE_MEMORY & UR_STABLE	N/A	Function dispose()	Abstract architecture does not consider closing the game, but the game requires state memory to be cleaned this is done using the function dispose() which disposes of all loaded textures and sprite batches
FR_REAL_TIME	N/A	Float gameTime	The float gameTime increments as the game runs using the time between frames, this is halted when the game is paused and resumes when game is unpaused
FR_RANDOMISED & UR_RANDOM	N/A	random(), rand.nextInt(mapWidth) rand.nextInt(mapHeight)	The concrete architecture considers random events which in this case is the generation of plunder in random locations by setting their location randomly on the map during initialization
FR_MAP NFR_DISPLAY NFR_GRAPHICS	N/A	private void initMap() private void drawMap()	The requirement for the map of the pirate game is to have a continuous map with a fixed layout, The function initMap generates a fixed layout of the map from the file located in the asset folder and the drawMap function draws the map tiles and also the colleges as well.