

# Architecture

Cohort 4: Group 2

## Greyhounds

Olwenn Gillespie

Glenn Asher

Praise Konji

Jennifer Oakley

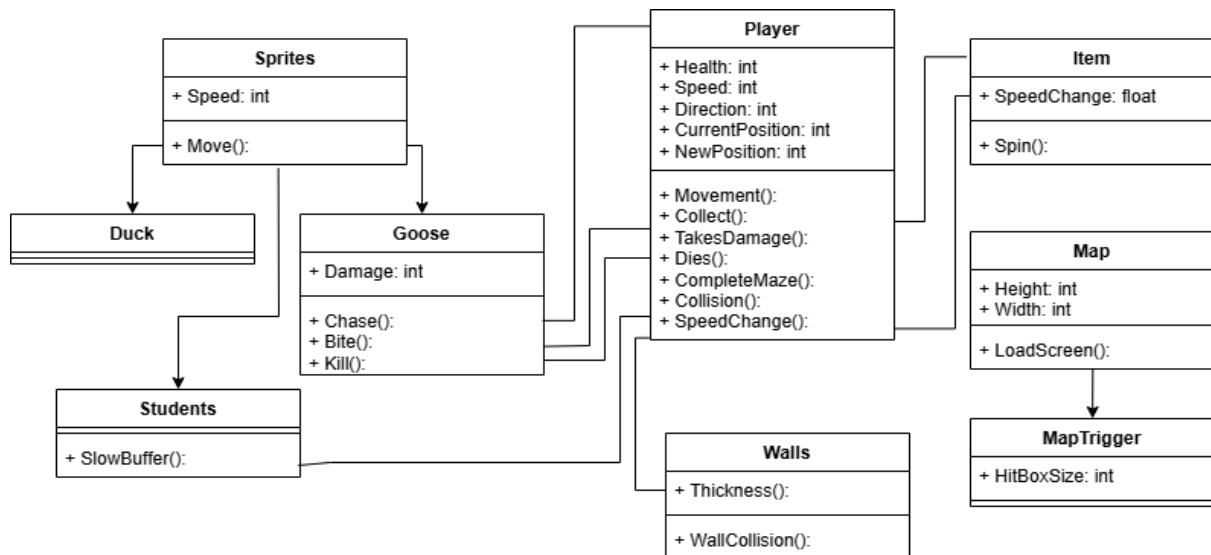
Alice Cavens

David Naji

Igor Dylewski

## Final Structural UML:

### Class UML:



Reference Table:

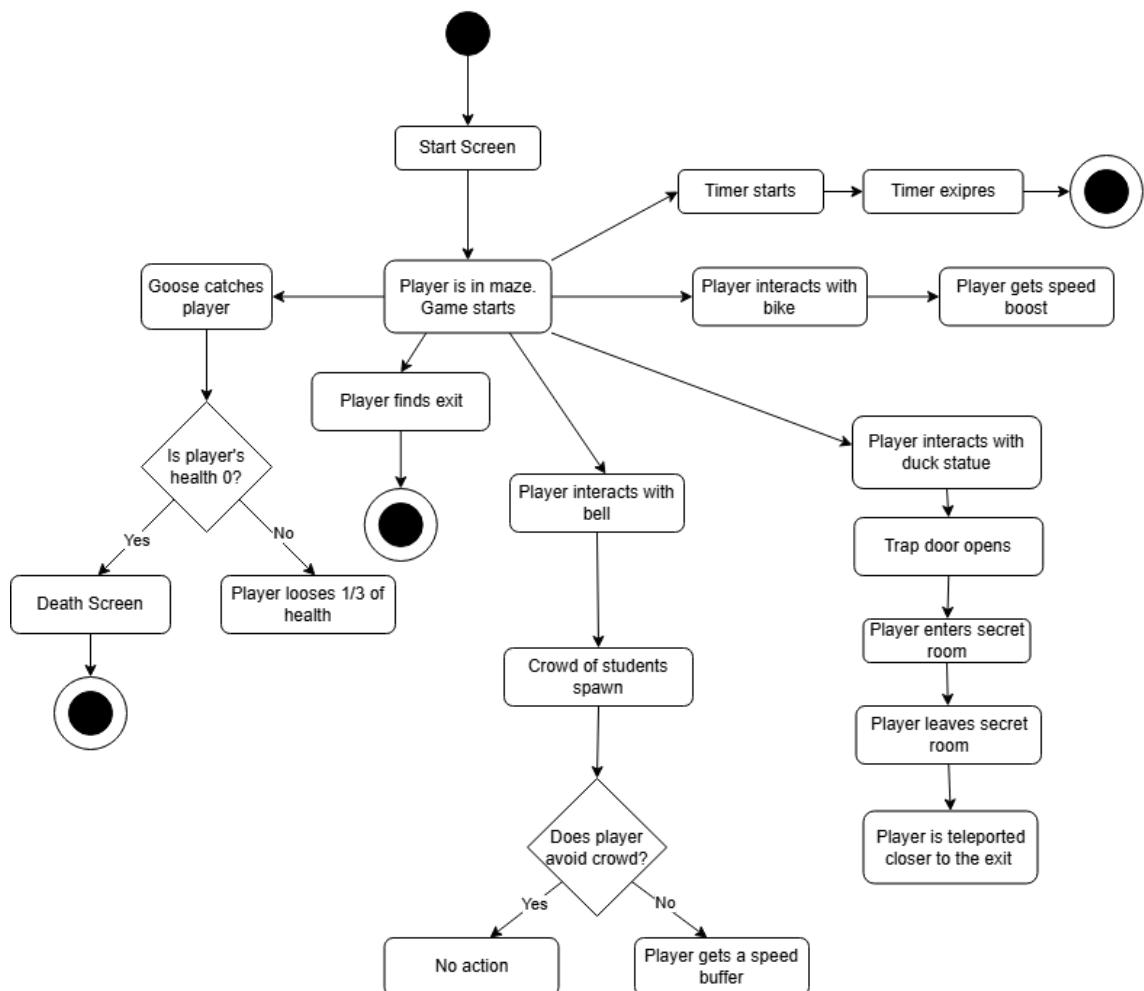
Class Attributes	Requirements ID
Goose: Chase()	FR_GOOSE_CHASE
Goose: Bite()	FR_GOOSE_BITE
Goose: Kill()	FR_LOSE
Students: SlowBuffer()	FR_SLOW_DOWN
Player: Movement()	UR_MOVE
Player: Collect()	FR_COUNTER
Player: TakesDamage()	FR_GOOSE_BITE
Player: Dies()	FR_LOSE
Player: CompleteMaze()	FR_WIN
Player: Collision()	FR_BOUNDARIES
Player: SpeedChange()	FR_SPEED_BOOST
Map: LoadScreen()	NFR_MAP_LOAD_TIME
MapTrigger	UR_RUN
WallCollision()	FR_BOUNDARIES

<b>Item</b>	<b>FR_SLOW_DOWN, FR_SPEED_BOOST</b>
-------------	---

For the structural diagrams, we wanted to use a UML class diagram. This type of diagram is useful for coding in Java, the class diagram shows the different classes and how they interact together, along with the attributes (functions) and operations (methods) belonging to each class. Inheritance can also be represented using the structure in the class diagram.

### Final Behavioural UMLs:

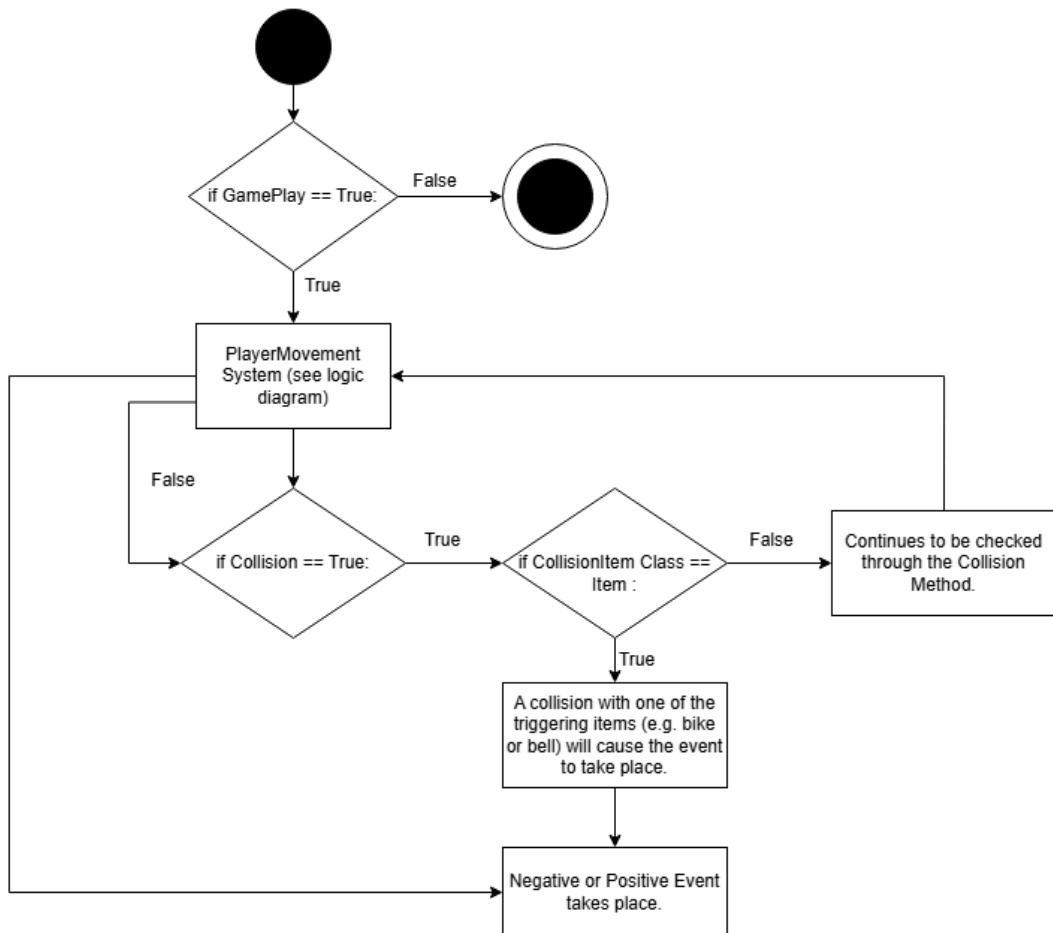
Activity Diagram:



<b>Event</b>	<b>Requirements ID</b>
<b>Start Screen</b>	<b>UR_RUN</b>
<b>Game Starts</b>	<b>FR_START_BUTTON</b>
<b>Timer starts</b>	<b>FR_TIMER</b>

Timer expires	<b>FR_TIME_OUT</b>
Player interacts with bike	<b>FR_SPEED BOOST</b>
Player interacts with bell	<b>FR_BELL</b>
Player gets slowed down	<b>FR_SLOW_DOWN</b>
Player interacts with duck statue	<b>FR_HIDDEN_EVENT</b>
Player loses 1/3 of health	<b>FR_GOOSE_BITE</b>
Death screen	<b>FR_LOSE</b>
Player finds exit	<b>FR_WIN</b>

### Event Logic Activity UML:



## Player Movement

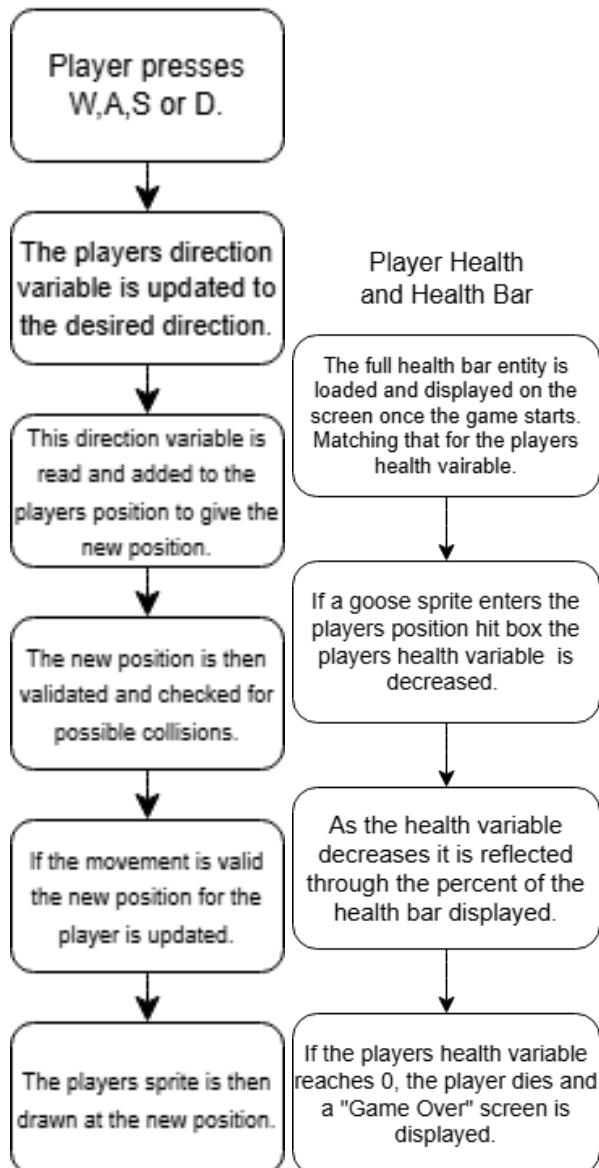


Diagram:	Requirements 1
<b>Player Movement System</b>	<b>FR_MOVEMENT</b>
Health bar	UR_HUD
Collision with goose sprite decreases health	FR_GOOSE_BITE
Game over	FR_LOSE

For the behavioural UMLs, we used activity diagrams as we thought it was good for visualising the flow of events in the game. It's also good for modelling in-game logic. We opted to do an activity diagram for the overall game behaviour of the whole game and a logic diagram for the events player movement and health bar.

We used [draw.io](#) to produce all the diagrams and Excel to produce the tables.

### Design Process:

Prior to the client meeting, we had several different ideas, including multiple mini mazes, a beer icon that slows you down for the negative event (relating to the university experience); having a collectible item like a student card (to open other areas of the map) for the hidden event. After discussing our ideas and the specification with the client, we decided that some of these ideas were not appropriate for the game or too complex for assignment 1, for example for the beer icon; the client wanted it to be playable for sixth form students. So we refined our current ideas and brainstormed new ones.

We introduced CRC cards to help us brainstorm more practical ideas for the game (these can be viewed on the website), we came up with: the negative event could be a speed buffer; the positive event could be a shield for the player to be protected from the dean the hidden event could be interacting with a duck statue (as the University of York has a statue commemorating LongBoi on campus) that could teleport you to an area closer to the exit. We also created an initial behavioural and structural UML to help visualize how the game would work, shown below.

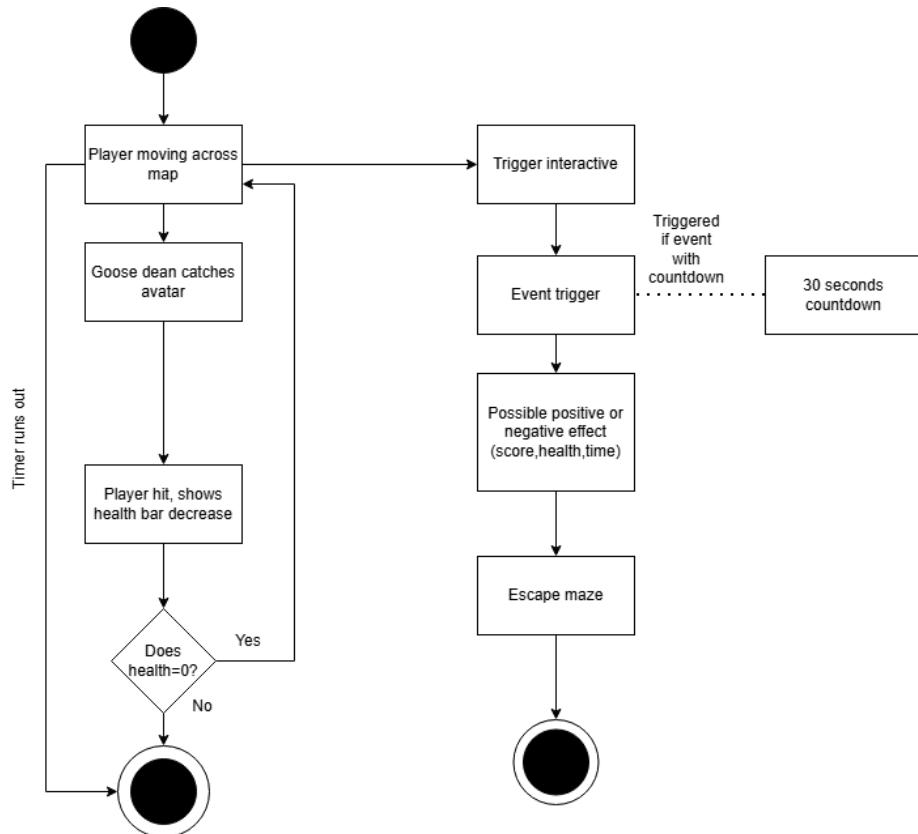
Eventually, we decided to expand the negative event to be more fun and unique to our game. The negative event would now be: the player interacts with the bell icon, a student crowd would appear, if the player collides with a student they get a slow movement buffer for a few seconds. We also changed the positive event to be a speed boost, as we thought it would be more fun than a shield. In addition to the event changes we wanted the dean to be a goose, as there are many geese on West Campus that are slightly scary. We also wanted to develop the hidden event by adding an underground area, with a duck cult to add an interesting underlying plot to the game, a goose vs. duck narrative.

To make the game have more of a university feel we decided to make the game look outdoors, but to scatter buildings that resemble the ones on campus around the map and to add in a lake like the one on West Campus. The duck statue will hopefully give the game a more University of York feel too.

The final behavioural and structural UMLs are shown above.

## Initial Designs:

### Initial Behaviour UML:



### Initial Class Diagram:

