

ENG1 Architecture

Cohort 3 - Group 28

“Team 28”

Muhammed Salahudheen

Joel McBride

Jamie Rogers

Maciek Zaweracz

Rhys Yeaxlee

Alex Spencer

Alex Firth

3. Architecture [22 marks]:

Give diagrammatic representations (structural and behavioural diagrams) of the architecture of the team’s product, with a brief statement of the specific languages (for instance, relevant parts of UML) and the tool(s) used to create these representations. Include a systematic justification for this architecture and describe how it was initially designed and how it evolved over the course of the project. Provide evidence of the design process followed (e.g. interim versions of architectural diagrams, CRC cards) on your team’s website and link to them from your report. Relate the architecture clearly to the requirements, using your requirements referencing for identification, and consistent naming of constructs to provide traceability (22 marks, ≤ 6 pages).

CRC CARDS



Candidates were created to closely follow the requirements, describing the intended function and listing the User Requirement within. Other team members clarified about how certain mechanics would work within the libgdx (i.e. how bounds should be handled and how movement should be thought about), resulting in the next stage of cards where there was no boundary candidate and a separate movement object was used to move objects like the avatar.

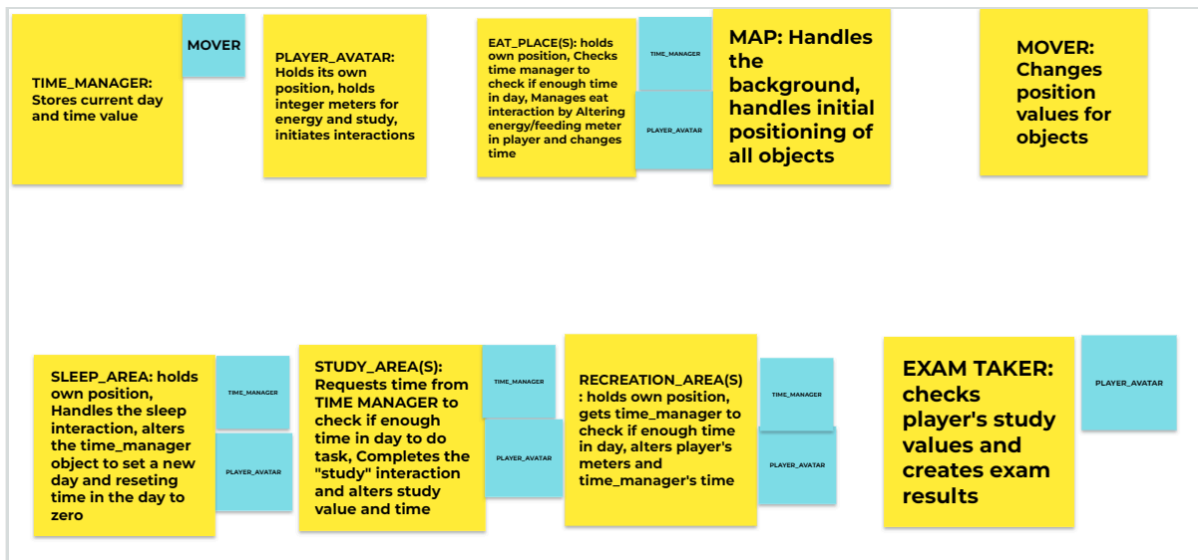
Stereotypes:



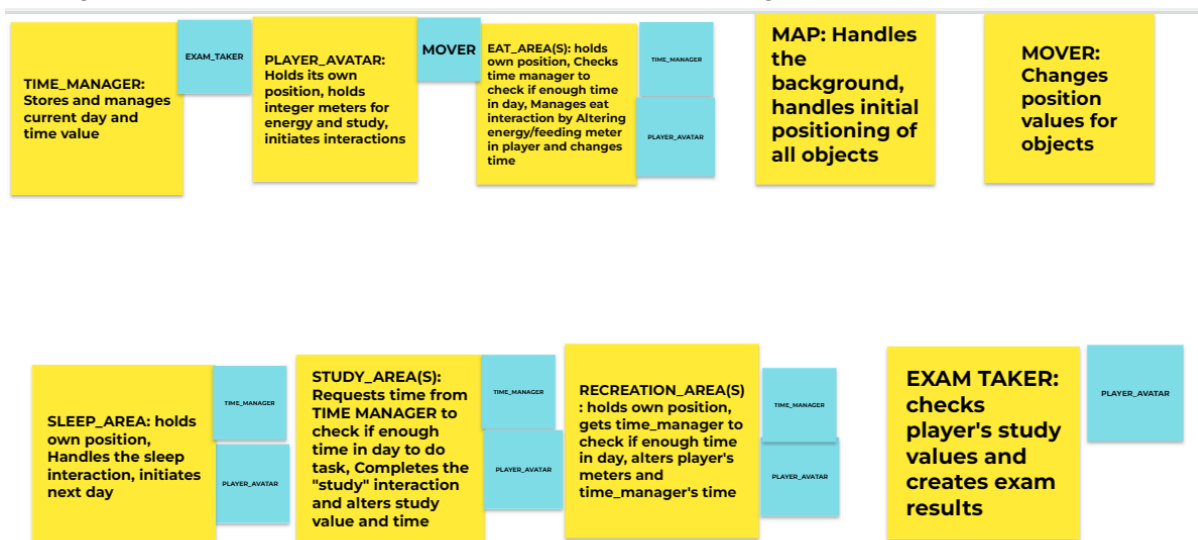
At this stage the stereotypes were relatively simple to assign. Feedback from teammates indicates that the study counters and energy meter should be part of the avatar object as they all relate directly to the player and there's no reason to have to fetch that between separate objects. Timer and day meter are also merged as they effectively operate as two parts of the same mechanic.



These simplified cards will be used as the basis for the next step of defining responsibilities. Names have been changed to reflect new roles or to better identify them.

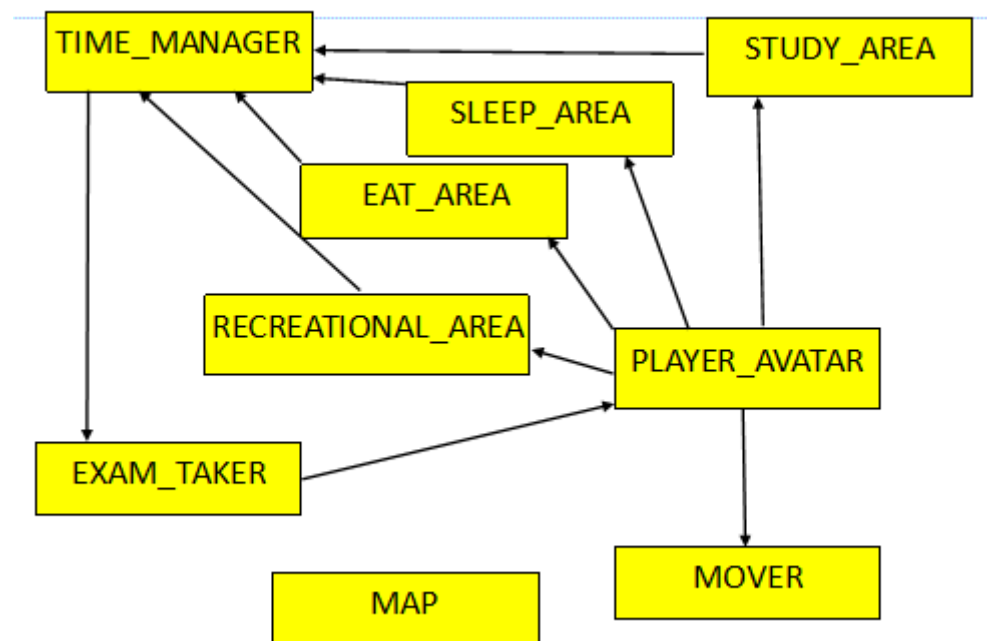


Adding responsibilities was simple as the connections in logic were not complex.



Upon further discussion with the team this was simplified to keep options open. The sleep area only initiates the day transfer, time_manager performs the specifics, meaning that it could be possible for other objects to transition days if needed.

Control Style



This basic logic map shows the logic flow at this stage. A dispersed control style is used as the game is too basic for a central smart object to be of much use.

CLASS TABLE

Class	Requirements	Implementation
GameManager	UR_IDEAL_LENGTH	Once timer reaches 10 min, do X
	UR_LENGTH	Once timer reaches 15 min, do X (force quit game?)
	UR_GAME_DURATION	
UIManager	UR_ACTIVITY_TRACKER	
	UR_ACTIVITY_COUNTER	
	UR_RESOURCE_CONSUMPTION	Visual Representation of resource consumption
MapManager	UR_MAP	- Takes care of building placement
	UR_BOUNDS	- Currently inside player,
Player	UR_GAME_AVATAR	- Private MoveComponent moveComponent = new MoveComponent(); - use this to move player

	UR_AVATAR_RECOGNISE	
	UR_RESOURCE_CONSUMPTION	Internal resource counters
MoveComponent	Part of UR_GAME_AVATAR	Allows an object to move
Interactable	-	<ul style="list-style-type: none"> - Private float interactRange; - Protected Void Interact(object_interacting);
StudyArea implements Interactable	UR_STUDY_AREA	
SleepArea implements Interactable	UR_SLEEP_AREA	
RecreationalArea implements Interactable	UR_RECREATIONAL_AREA	
EatArea implements Interactable	UR_EAT_AREA	
Exam Manager	UR_Exam	?