22 | R1M1

November 6, 2022

Berfan Pirhan | Department of Computer Engineering - Koç University
Hakan Çapuk | Department of Computer Engineering - Koç University
Erim Satar | Department of Computer Engineering - Koç University
Eren Ceylan | Department of Computer Engineering - Koç University
Mehmet Bartu Celasun | Department of Computer Engineering - Koç University

INDEX

Use Case Diagram

Use Cases

UC1 : Authorize Player (Berfan)
UC2 : Pause/Resume Game (Bartu)
UC3 : Exit Game (Erim, Hakan)
UC4 : Build Map (Eren)
UC5 : Play Game (Eren)
UC6 : Spawn Aliens (Bartu)
UC7 : Spawn/Despawn Power-ups (Berfan)
UC8 : Move Character (Erim, Hakan)
UC9 : Collect Items (Hakan ,Erim)
UC10 : Use Power-ups (Hakan, Erim)
UC11 : Use Key (Erim, Hakan)
UC12: Exit Application (Hakan)
UC13: Open Help Screen (Erim)
UC14: Randomize Object Locations (Eren)

Use Case Narratives

Use Case UC1 — Authorize Player

Use Case Name	Authorize player
Scope	Login screen
Level	Subfunction

Primary Actor	Player
Stakeholders & Interests	- Player: Wants to successfully register themselves and log in to the game.
Preconditions	- The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation from a mouse. - The player must have launched the executable for the game. - The player must have a valid email address to enter to register to the game database.
Success Guarantee	The login menu is launched with its respective buttons and textboxes in a distinct and visible form. The player can enter their email and a unique username and register, and then enter that username with their email to log in and start playing the game.
Main Success Scenario	1. The player launches the game. 2. The player enters a valid email address and a username of their choice, and registers for the game. 3. The player enters their unique username with their email to log in. 4. The player logs in to the game successfully.
Extensions	2a. The player enters an invalid email address. 1. The player is prompted with an error pop-up window, telling them to enter a valid email address. 2b. The player enters a username that already exists on the database to register to the game. 1. The player is prompted with an error pop-up window, telling them to enter a different username. 3a. The player enters a username different from the unique one they used to register. 1. The player is prompted with an error pop-up window, telling them to enter the username they used to register for the game. 3b. The player tries to log in without registering into the database with a valid email and username. 1. The player is prompted with an error pop-up window, telling them to enter a valid email address and a unique username to register first. 3c. The player tries to input a username with the incorrect email address to log in. 1. The player is prompted with an error pop-up window, telling them to enter the email they used with their username to log in.
Special Requirements	- The textbox that the user will write their name in should be large enough to see the text being inputted, as well as the buttons to register these

	inputs. - The buttons for receiving the given inputs and the textboxes should be distinctly placed on the login screen. - The buttons should explicitly display their functionality.
Technology & Data Requirements List	a. The device must be able to receive keyboard inputs and left-click inputs from a mouse.
Frequency of Occurrence	Any time when the player launches the game.
Miscellaneous	- No open issues.

Use Case UC2 — Pause/Resume Game

Use Case Name	Pause/Resume Game
Scope	Running Mode
Level	Subfunctional
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to pause the game to take a break and resume it to play the game again.
Preconditions	- The player must be successfully logged in. - The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation of a mouse. - The player must be in running mode.
Success Guarantee	The game and all animations stop once the player presses the pause button. The player will not be able to perform any action until they press the resume button. The player can play the game again from where they had left off once they press the resume button.
Main Success Scenario	1. The player presses the escape button while they are in running mode. 2. The game stops and restricts all character actions. 3. The pause menu shows up. 4. The player clicks the resume button and the game resumes again.
Extensions	4a. The player clicks on the help screen on the pause menu before resuming the game. 1. The help screen shows up with information regarding game mechanics and objects. 2. The game session remains paused.

Special Requirements	- The pause menu should be clearly visible along with the different options/buttons in it.
Technology & Data Requirements List	a. The device must be able to receive keyboard inputs and left-click inputs from a mouse.
Frequency of Occurrence	Any time when the player presses the escape button in running mode.
Miscellaneous	No open issues.

Use Case UC3 — Exit Game

Use Case Name	Exit Game
Scope	Running Mode
Level	Subfunctional
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to end the current game session and return to the main menu.
Preconditions	- The player must be successfully logged in. - The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation of a mouse. - The player must create a map using Building Mode according to the map rules. - The player must have started Running Mode. - The player must have paused the game.
Success Guarantee	The player can press the pause button to get the pause menu to open. From the pause menu, the player can click the exit button to exit from the game session back to the main menu.
Main Success Scenario	1. The player pauses the game and the pause menu appears. 2. The player clicks the exit button on the pause menu. 3. The game session ends and the player returns to the main menu.
Extensions	No extensions.
Special Requirements	- The exit button should be clearly visible.
Technology & Data	a. The device must be able to receive the arrow

Technology & Data Requirements List	a. The device must be able to receive the arrow inputs from a keyboard and left-click inputs from a mouse.
Frequency of Occurrence	Any time when the player wants to exit the current game session.
Miscellaneous	No open issues.

Use Case UC4 — Build Map

Use Case Name	Build Map
Scope	Building Mode
Level	User-goal
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to place various objects into different buildings (“Student Center”, “CASE Building”, “SOS Building”, “SCI Building”, “ENG Building”, “SNA Building”) to set the map.
Preconditions	- The player must be successfully logged in. - The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation of a mouse.
Success Guarantee	The placement of the objects and their numbers based on the buildings are saved. The map is created for Running Mode and the objects to hold the keys are successfully picked.
Main Success Scenario	1. The grid view of the map is created. The names of the buildings are shown for the player and a separate grid object is created for showing/adding the objects to the buildings. 2. Instructions for this process are shown as a small closable pop-up. 3. The current building map and the small icons of the objects are shown to the player. 4. The player left-clicks on an object icon. 5. The player clicks on the desired location on the map and places the object there. 6. The player moves to the next building. *The player repeats steps 3-6 until each building has the minimum number of required objects.* 7. The player is asked if they are sure with their placement. 8. The player finishes building the map and exits

	Building Mode.
Extensions	2a. The player does not press the close button for the instructions. 1. The map and the objects are not shown until the instructions are closed. 5a. The player tries to place two or more objects that collide with each other. 1. The placement of the object is not accepted, and the object is not placed on that location. 2. The player gets a collision error. 6a. The player tries to move to the next room without satisfying the required object count in their current room. 1. The player gets a pop-up error telling them to place X more objects, and the player cannot move on to the next building. 1a. The player does not close the pop-up error window. 1. The game freezes until the window is closed. 7a. The player picks the option to verify that they are sure about the object count. 1. The game proceeds to Running Mode. 7b. The player picks the option to add more objects to the grid world. 1. The player teleports to the Student Center and places extra objects to any room they want, with all objects still in place. 7c. The player does not pick any options. 1. The game freezes until the player clicks on an option.
Special Requirements	- The grid world must show the buildings and their names clearly. - The size of each building grid must be at least 25 times larger than each object/furniture. - The objects/furniture should all fit in one square grid.
Technology & Data Requirements List	a. The device must be able to receive the arrow inputs from a keyboard and left-click inputs from a mouse.
Frequency of Occurrence	Once at the beginning of every game session.
Miscellaneous	No open issues.

Use Case UC5 — Play Game

Use Case Name	Play Game
Scope	Running Mode

Level	User-goal
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to run through the current building to find the key and do this for all rooms until they find all six keys and escape from the school.
Preconditions	- The player must be successfully logged in. - The player must successfully finish the game's Building Mode. - The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation from a mouse.
Success Guarantee	The grid world is created from the data collected in Building Mode. All objects are placed without any collision. A single key is hidden in an object in each building. The lives of the player, the remaining time in each building and the power-ups in the player's bag are clearly displayed and dynamically updated. The power-ups and aliens spawn and function with their intended purpose.
Main Success Scenario	1. The player spawns in the first building in a random location, and the game starts. 2. The player plays the game in each building in accordance with the character control mechanics, power-up mechanics, key collection mechanics, and avoiding aliens. 3. The player moves on to the next building after collecting the key for the current building they are in. *The player repeats steps 2-3 until they collect all six keys and successfully escape the school.* 4. The player collects all six keys and escapes from the school, and the game terminates.
Extensions	2a. The player interacts with a power-up. 1. The character uses it in accordance with the Collect Item and Use Power-up use cases. 2b. The player finds the key hidden in the building. 1. The character utilizes it in accordance with the Collect Item and Use Key use cases. 2c. The player gets in the range of a Shooter Alien. 1. The player loses a life. 2d. The player comes in contact with a Blind Alien. 1. The player loses all of their lives. The game terminates and gives a "You lost!" or "Game Over" message to the user, and the Game Over screen pops up. 2e. The player interacts with a Time-wasting Alien. 1. Nothing happens. 2f. The player loses all of their lives. 1. The game terminates and gives a "You lost!" or

	“Game Over” message to the user, and the Game Over screen pops up. 2g. The player runs out of time in the building they are in. 1. The game terminates and gives a “You lost!” or “Game Over” message to the user, and the Game Over screen pops up. 2h. The player restarts the game session after losing the game, using a restart button. 1. The game session starts again, with the predefined game layout and settings. 3a. The player tries to move on to the next building without finding the key in their current building. 1. An error message is displayed, warning the player that they have not found the key to escape that building. 3b. The player tries to move on to a building that has not yet been unlocked. 1. An error message is displayed, warning the player that they have not yet unlocked that building.
Special Requirements	- The grid world must clearly show the buildings along with the objects in them. - The size of each building grid must be at least 25 times bigger than the player’s icon/avatar, power-ups and aliens. - All objects, aliens and power-ups should be clearly distinguishable from one another.
Technology & Data Requirements List	a. The device must be able to receive the arrow inputs from a keyboard and left-click inputs from a mouse.
Frequency of Occurrence	Once after Building Mode is finished.
Miscellaneous	No open issues.

Use Case UC6 — Spawn Alien

Use Case Name	Spawn Alien
Scope	Running Mode
Level	Subfunctional
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to play the game in a more challenging and interactive way with the inclusion of hostile entities.
Preconditions	- The player must be successfully logged in. - The player must be in Running Mode.

	- The buildings must have enough space to spawn aliens.
Success Guarantee	An alien spawns within the current building the player is in as the player plays the game. All aliens act in accordance with their type.
Main Success Scenario	1. The player enters a building. 2. After the player enters a building, a random alien spawns in that building every 10 seconds. *Step 2 repeats until the player finds the key in the current building and escapes from that building, and also if there is enough space in the building.*
Extensions	2a. The alien that spawns is a “Shooter Alien”. 1. A “Shooter Alien” spawns in the building and starts shooting in a direction every second. 2b. The alien that spawns is a “Blind Alien”. 1. A “Blind Alien” spawns in the building and starts moving around in random directions. 2c. The alien that spawns is a “Time-wasting Alien”. 1. A “Time-wasting Alien” spawns in the building and changes the key’s location every 5 seconds. 2d. The building does not have enough space left to spawn an alien. 1. New aliens stop spawning in that building.
Special Requirements	- The alien that spawns in a building should be clearly distinguishable from other objects. - The alien cannot perform its ability in the first 2 seconds after it spawns. - The alien that spawns should fit in one square grid, having equal size with power-ups and other objects.
Technology & Data Requirements List	a. The device must be able to receive keyboard inputs and left-click inputs from a mouse.
Frequency of Occurrence	Every 10 seconds in a building.
Miscellaneous	No open issues.

Use Case UC7 — Spawn/Despawn Power-up

Use Case Name	Spawn/Despawn Power-up
Scope	Running Mode
Level	Subfunctional
Primary Actor	Player

Stakeholders & Interests	- Player: Wants to use power-ups to escape from the aliens and find the keys easily.
Preconditions	- The player must be successfully logged in to the game. - The game must be in Running Mode. - The rooms must have enough space for the power-ups to be successfully spawned.
Success Guarantee	The power-ups spawn in distinct and random locations in the rooms every 12 seconds, and vanish in 6 seconds if not collected. Five types of power-ups can be spawned (extra life, extra time, protection vest, plastic bottle, hint). The player can collect the power-ups if they desire with the right-click inputs of a mouse.
Main Success Scenario	1. The power-ups randomly spawn in the room the player is in. 2. The player collects the power-up and utilizes it in accordance with the use cases for collecting items and using power-ups.
Extensions	1a. The power-up that spawns is an extra life power-up. 1. The player can right-click on the power-up to get an extra life (further details in the Use Power-up use case). 1b. The power-up that spawns is an extra time power-up. 1. The player can right-click on the power-up to get more time (further details in the Use Power-up use case). 1c. The power-up that spawns is a protection vest power-up. 1. The player can right-click on the power-up to obtain protection (further details in the Use Power-up use case). 1d. The power-up that spawns is a hint power-up. 1. The player can right-click on the power-up to get the key's location (further details in the Use Power-up use case). 1e. The power-up that spawns is a plastic bottle power-up. 1. The player can right-click on the power-up to get a bottle to distract a Blind Alien. (further details in the Use Power-up use case). 2a. The player does not collect a power-up that spawns. 1. The power-up disappears in 6 seconds after spawning and not being collected.
Special Requirements	- The power-ups should be clearly distinguishable from the other objects that spawn in each room. - The bag that will contain the collectible power-ups should be clearly visible along with the power-ups themselves. - The power-up that spawns should fit in one square grid.
Technology & Data Requirements List	a. The device must be able to receive keyboard inputs and right-click inputs from a mouse.

Frequency of Occurrence	Every 12 seconds in random locations within the building the player is in.
Miscellaneous	No open issues.

Use Case UC8 — Move Player

Use Case Name	Move Player
Scope	Running Mode
Level	User-goal
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to change the position of the player to play the game.
Preconditions	- The player must be successfully logged in. - The player must be using a device that can receive keyboard inputs. - The player must have created a map in Building Mode according to the map rules. - The player must be in Running Mode.
Success Guarantee	The player can move around the map successfully using the arrow keys.
Main Success Scenario	1. The player presses the arrow keys to move the character. 2. The player successfully changes position based on the direction they press (north, south, east, west).
Extensions	1a. The player presses the up arrow key on their keyboard. 1. The character moves to the north. 1b. The player presses the down arrow key on their keyboard. 1. The character moves to the south. 1c. The player presses the right arrow key on their keyboard. 1. The character moves to the east. 1d. The player presses the left arrow key on their keyboard. 1. The character moves to the west. 2a. The player tries to move through an object/furniture or a wall. 1. The character stays still.
Special Requirements	- The playable character and the game objects should be clearly visible and distinguishable from

	one another to prevent visibility problems in the game.
Technology & Data Requirements List	a. The device must be able to receive arrow inputs from a keyboard and both left-click and right-click inputs from a mouse.
Frequency of Occurrence	Any time when the player presses the arrow keys on their keyboard.
Miscellaneous	No open issues.

Use Case UC9 — Collect Item

Use Case Name	Collect Item
Scope	Running Mode
Level	User-goal
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to collect items and either use them on the spot or store them in their bag while they play the game.
Preconditions	- The player must be successfully logged in. - The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation of a mouse. - The player must have created a map in Building Mode according to the map rules. - The player must be in Running Mode.
Success Guarantee	The player can collect power-ups or keys, and either use them on the spot or store them in their inventory for later use. The “extra life” and “extra time” power-ups are used the moment they are collected. The other power-ups are stored in the player’s inventory/bag if desired. The collected key is stored in the bag.
Main Success Scenario	1. The player right-clicks their mouse to collect the power-ups from any distance, or collects a key by left-clicking on it when they are close enough to it. The player stores the collectible items if they plan to use them later. 2. The player successfully collects and/or utilizes the item.
Extensions	1a. The player right-clicks on the extra life or extra time power-ups. 1. The power-up is used immediately and its effect takes place

	takes place. 1b. The player right-clicks on the hint, protection vest or plastic bottle power-up. 1. The player is prompted with a pop-up screen telling them if they want to store the item in their bag. 1a. The player chooses to use the power-up immediately. 1. The power-up is used and its effect takes place. 1b. The player chooses to store the power-up in their inventory. 1. The power-up is stored in the bag and appears in the player's inventory. 1c. The player left-clicks on an object that does not contain the key. 1. Nothing happens. 1d. The player left-clicks on an object that contains the key, without being close enough to it. 1. Nothing happens. 1e. The player left-clicks on an object that has the key, while being close enough to it. 1. The player collects the key and it is stored in their inventory.
Special Requirements	- The playable character, collectible items and the inventory/bag should be distinguishable to not cause visibility problems while playing the game.
Technology & Data Requirements List	a. The device must be able to receive keyboard inputs and both left-click and right-click inputs of a mouse.
Frequency of Occurrence	Any time when the player left-clicks or right-clicks based on the item they want to interact with.
Miscellaneous	No open issues.

Use Case UC10 — Use Power-up

Use Case Name	Use Power-up
Scope	Running Mode
Level	User-goal
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to use power-ups to help avoid the aliens and escape the buildings easily.
Preconditions	- The player must be successfully logged in. - The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation of a mouse.

	operation of a mouse. - The player must have created a map in Building Mode according to the map rules. - The player must be in Running Mode.
Success Guarantee	The player uses a power-up and obtains its effects based on its type.
Main Success Scenario	1. The player uses the power-up they collected immediately, or uses one they have in their bag with the respective key inputs required. 2. The player obtains the effects of the power-up.
Extensions	1a. The player uses an extra time power-up. 1. An extra 5 seconds are added to the player's timer. 1b. The player uses an extra life power-up. 1. An extra life is added to the player's lives. 1c. The player uses a hint power-up either directly, or by pressing the "H" button on their keyboard to use it from their bag. 1. A small 4x4 rectangle appears that highlights the key's location for 10 seconds. 1d. The player uses a protection vest power-up either directly, or by pressing the "P" button on their keyboard to use it from their bag. 1. The player is protected from the Shooter Alien for 20 seconds. 1e. The player uses a plastic bottle power-up either directly, or by pressing the "B" button on their keyboard to use it from their bag. 1. The player throws the bottle in a certain direction by using the A, D, W, X buttons on the keyboard to trick the Blind Alien. (A: west, D: east, W: north, X: south) 1a. The player does not press the A, D, W, or X buttons to throw the bottle. 1. Nothing happens. 1f. The player tries using a collectible item without having it in their inventory. 1. Nothing happens.
Special Requirements	- The bag and items in the bag should be distinguishable to not cause visibility problems while playing the game.
Technology & Data Requirements List	a. The device must be able to receive keyboard inputs and left-click inputs of a mouse.
Frequency of Occurrence	Any time when the player wants to use power-ups.
Miscellaneous	No open issues.

Use Case UC11 — Use Key

Use Case Name	Use Key
Scope	Running Mode
Level	User-goal
Primary Actor	Player
Stakeholders & Interests	- Player : Wants to use the keys to open the doors of the buildings and escape from them.
Preconditions	- The player must be successfully logged in. - The player must be using a device that can receive keyboard inputs. - The player must be able to use the click operation of a mouse. - The player must have created a map in Building Mode according to the map rules. - The player must be in Running Mode.
Success Guarantee	The player finds a key in one of the objects in the building they are in, the key appears for a second and the door is unlocked. The player can move on to the next building.
Main Success Scenario	1. The player uses the key they found hidden in a close object, by clicking on the key in their bag when they are close by the door. 2. The door of the building is unlocked.
Extensions	1a. The player is not close enough to a door to use the key. 1. Nothing happens. 1b. The player tries to unlock the door without having the key. 1. The door does not open. 1c. The player opens the door that exits from the last building. 1. The game finishes, and the game over screen pops up.
Special Requirements	- When the key is correctly used, it appears for a second, and opens the door. - The bag and items in the bag should be distinguishable to not cause visibility problems while playing the game. - There is a game-over screen when the game is finished, or when the player dies.
Technology & Data Requirements List	a. The device must be able to receive keyboard inputs and left-click inputs of a mouse.
Frequency of Occurrence	Any time when the player wants to use a key.

Miscellaneous	No open issues.

Additional Use Cases

Use Case UC12 - Exit Application
Player closes the Game App completely, all threads related to the game terminate, and the game window closes.

Use Case UC13 - Open Help Screen
Players can open the help menu in the home screen and pause screen. In the help screen, the player can see information about how to play the game, and information about the objects/furniture, aliens and power-ups.

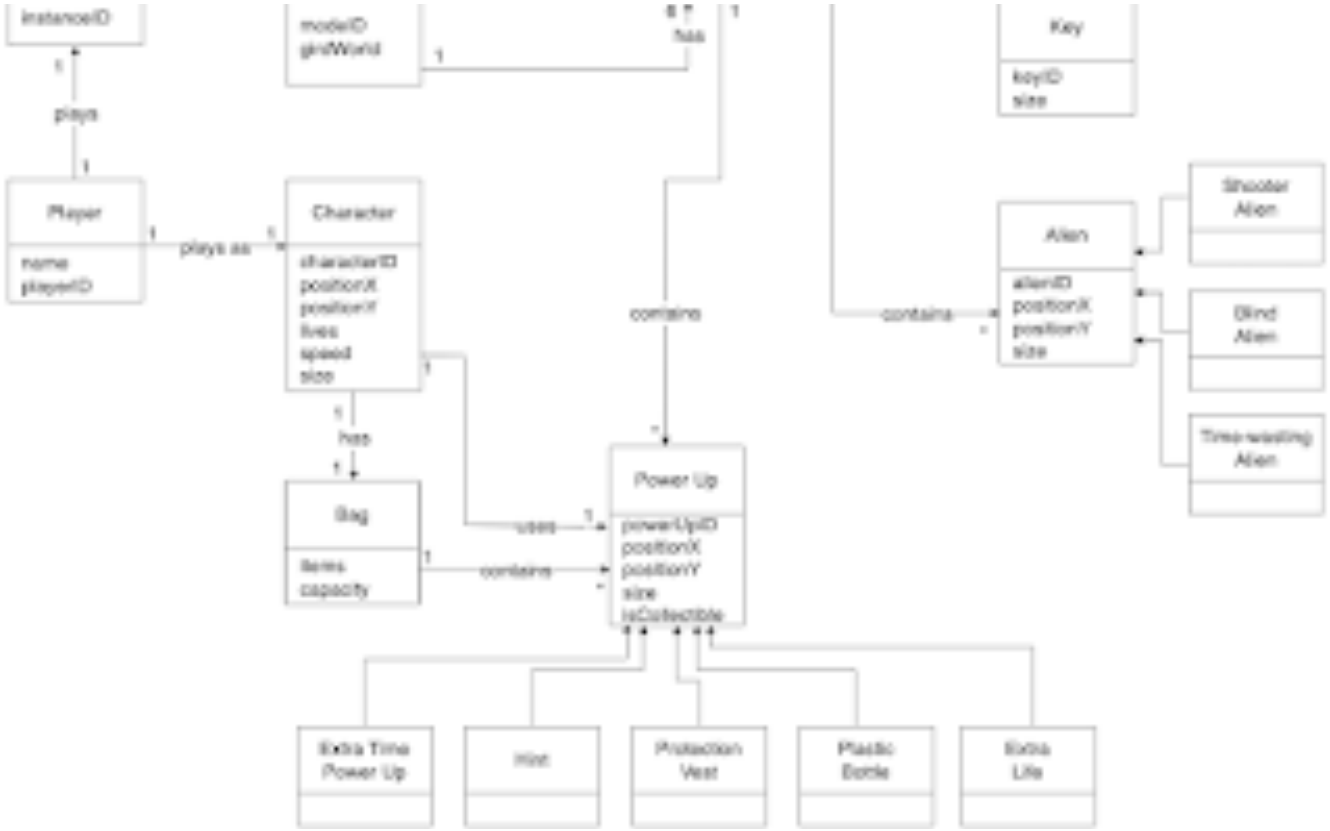
Use Case UC14 - Randomize Object Locations
When creating a new grid building, the objects in the building that the player placed in the building mode, will be randomly located.

Version History | Use Cases

Version	Date	Description	Author(s)
Draft v1	October 25, 2022	Use case separation of 9 use cases.	**Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk**
Draft v2	October 27, 2022	Detailing and formatting of 9 use cases.	**Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk**
Draft v3	November 1, 2022	Adding a 10th use case.	**Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk**
Revision	November 2, 2022	Modifying details of various use cases and fixing other related issues.	**Berfan Pirhan**

Revision	November 2, 2022	Divided collect Item and control Character use cases	**Erim Satar, Hakan Çapuk**
Revision	November 5, 2022	Updated and revised various use cases.	**Berfan Pirhan**
Draft v4	November 5, 2022	Added extra use cases without narratives.	**Eren Ceylan, Hakan Çapuk, Erim Satar**
Final	November 6, 2022	Finalized the use cases, narratives and their details.	**Berfan Pirhan**

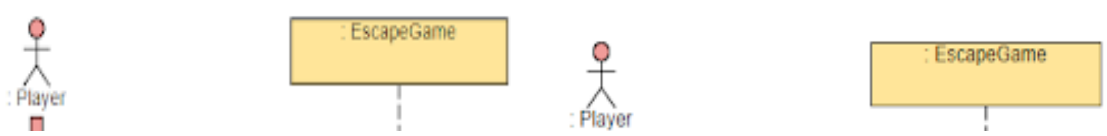
Domain Model Diagram

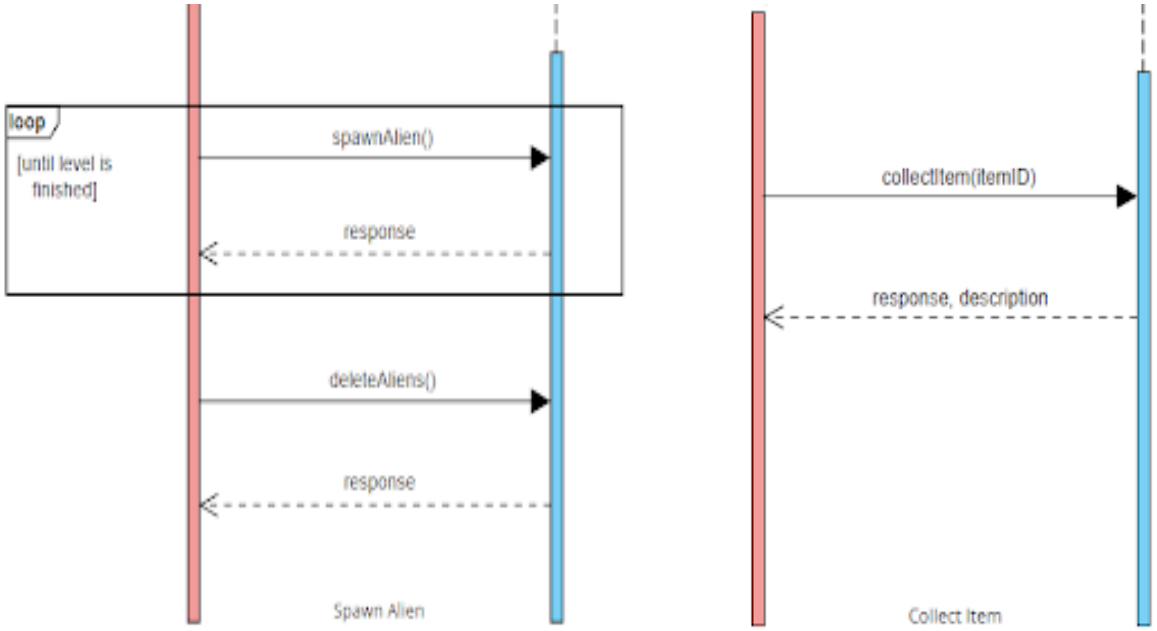


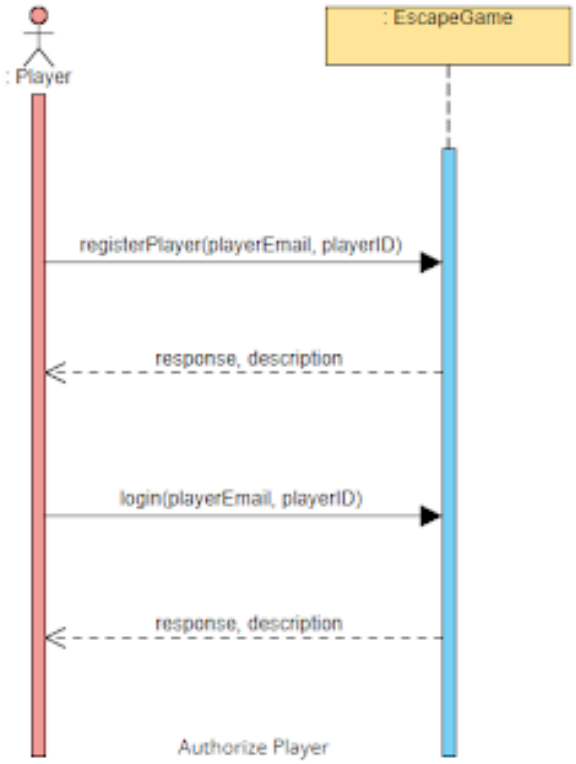
Version History | Domain Model Diagram

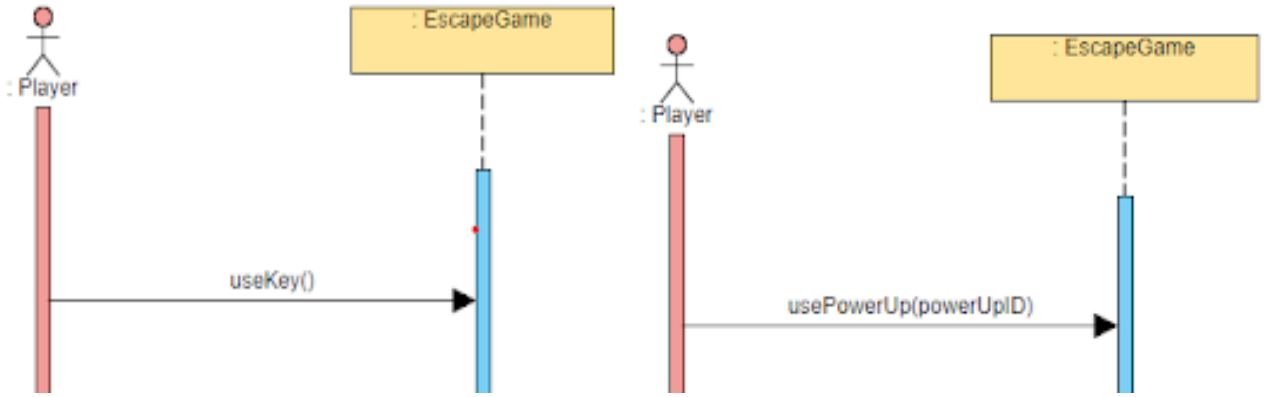
Version	Date	Description	Author(s)
Draft v1	November 3, 2022	Created the domain model diagram with various domain classes.	Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk
Final	November 5, 2022	Revised	Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk

System Sequence Diagrams

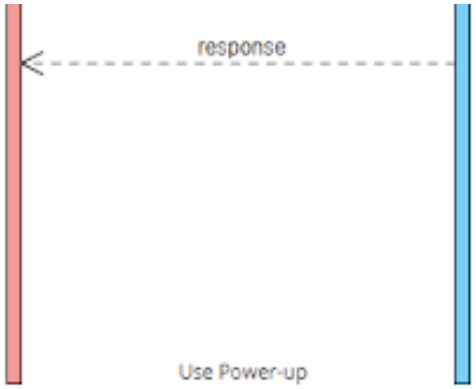


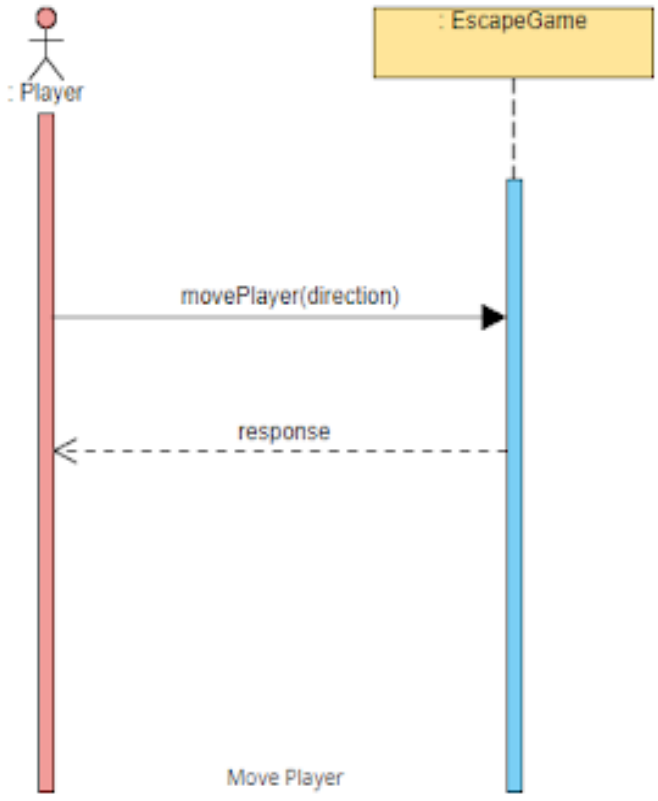

loop
[until level is finished]
spawnAlien()
response
deleteAliens()
response
Spawn Alien
collectItem(itemID)
response, description
Collect Item

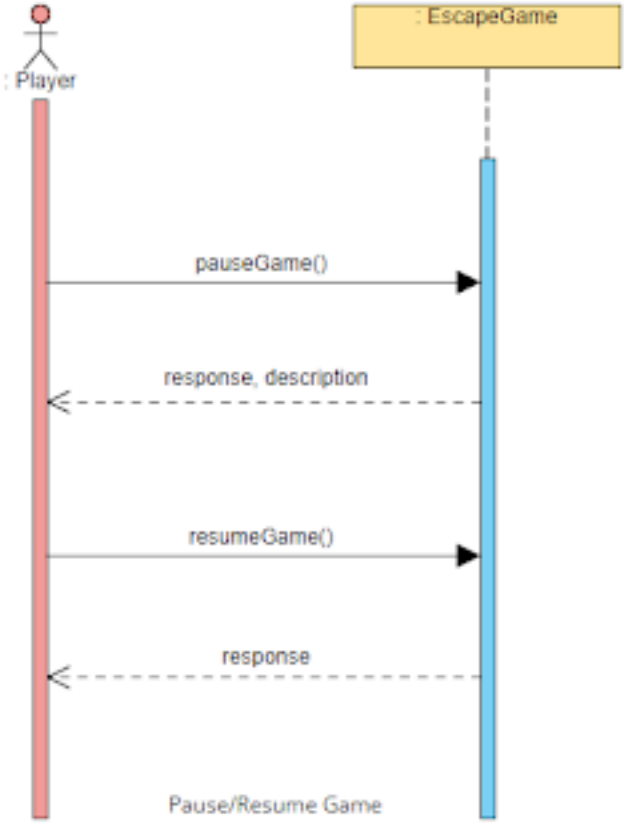

: Player
: EscapeGame
registerPlayer(playerEmail, playerID)
response, description
login(playerEmail, playerID)
response, description
Authorize Player

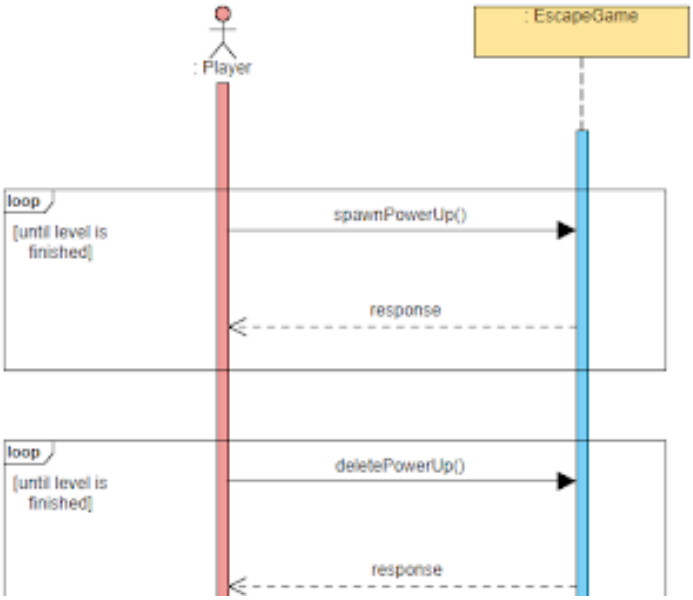

: Player
: EscapeGame
useKey()
: Player
: EscapeGame
usePowerUp(powerUpID)

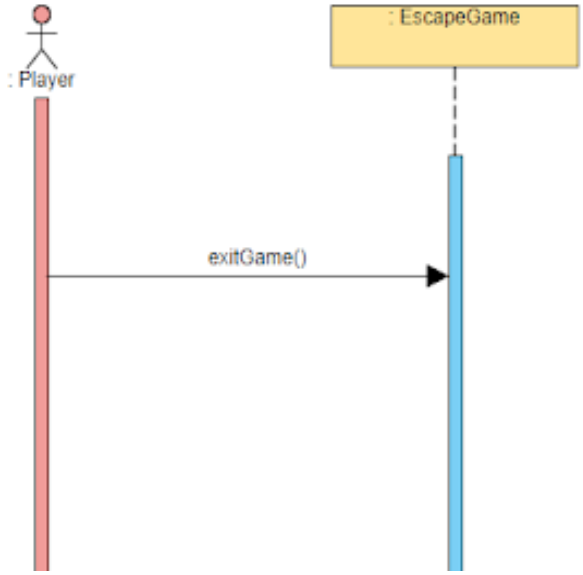

response
Use Key

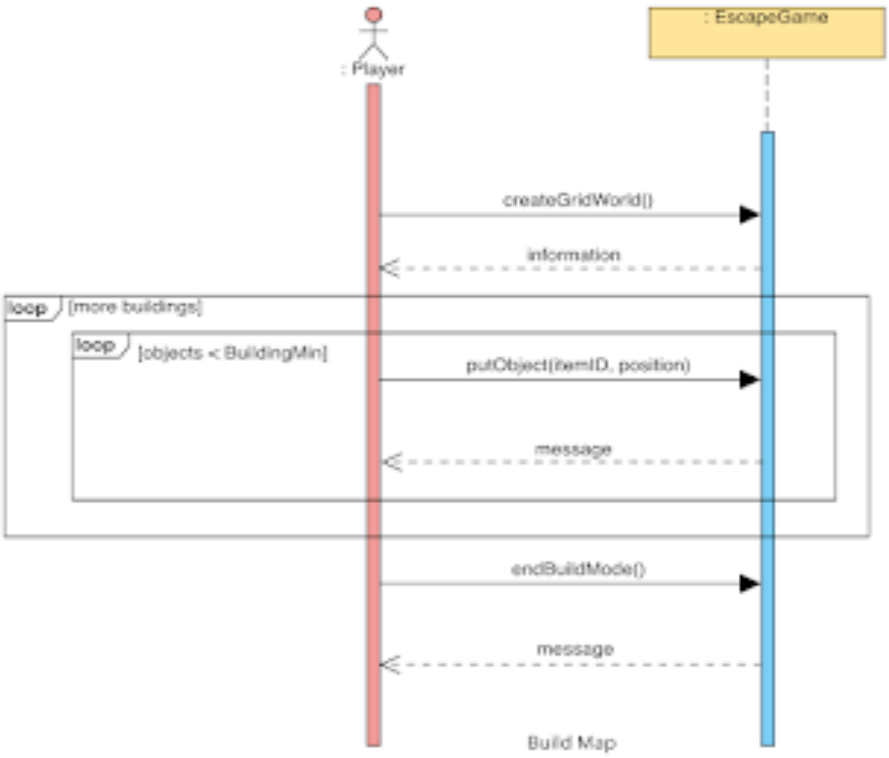

response
Use Power-up


: Player
: EscapeGame
movePlayer(direction)
response
Move Player

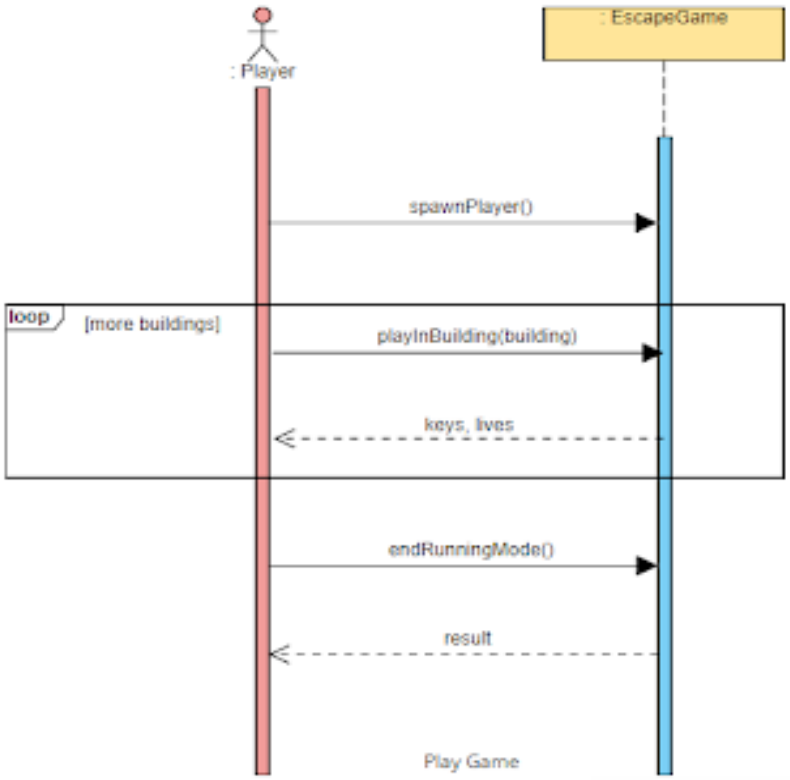

: Player
: EscapeGame
pauseGame()
response, description
resumeGame()
response
Pause/Resume Game


: Player
: EscapeGame
loop
[until level is finished]
spawnPowerUp()
response
loop
[until level is finished]
deletePowerUp()
response


: Player
: EscapeGame
exitGame()



Build Map



Play Game

Version History | System Sequence Diagrams

Version	Date	Description	Author(s)
Draft v1	November 2, 2022	Created drafts for the system sequence diagrams for most use case narratives.	**Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk**
Revision	November 3, 2022	Updated and polished the SSDs.	**Berfan Pirhan**
Revision	November 5, 2022	Fixed errors with some SSDs, and added SSD for “Authorize Player”.	**Berfan Pirhan**
Revision	November 5, 2022	Fixed Build Map SSD according to TA’s comment in the meeting	**Eren Ceylan**

Operation Contracts

Contract CO1 — registerPlayer

Operation:	**registerPlayer(playerEmail: String, playerID: String)**
Cross References:	**Authorize Player**
Preconditions:	**- The player must have launched the game executable.**
	- The player must have a valid email.
Postconditions:	**- A player instance was created, and added to the game database. (instance creation)**
	- The player has been associated with the given email and username. (association formed)

Contract CO2 — login

Operation:	**login(playerEmail: String, playerID: String)**
Cross References:	**Authorize Player**
Preconditions:	**- The player must have launched the game executable.**
	- The player must have registered with a unique username.

	- The player must use the email used for the username.
Postconditions:	- The player is associated with the game after logging in. (association formed) - The instance ID attribute of the game instance is updated with the main menu screen. (attribute modification)

Contract CO3 — exitGame

Operation:	exitGame()
Cross References:	Exit Game
Preconditions:	- The player is in Running Mode. - The player is on the pause menu.
Postconditions:	- The current running mode instance is deleted, along with all other instances contained in it. (instance deletion) - The instance ID of the game is updated with the main menu screen. (attribute modification).

Contract CO4 — createGridWorld

Operation:	createGridWorld()
Cross References:	Build Map
Preconditions:	- The player has pressed the play button for the game on the main menu.
Postconditions:	- A new Building Mode instance is created. (instance creation) - Six building instances are created. (instance creation) - Each building instance is associated with the grid world attribute of the Building Mode instance. (association formed) - The grid world attribute of the building mode instance is dynamically updated with each object placed. (attribute modification).

Contract CO5 — putObject

Operation:	putObject(itemID: String, position: Point)
Cross References:	Build Map
Preconditions:	- The player must have started Building Mode. The player must not have exceeded the object limit for the building

	the building.
Postconditions:	**- An object/furniture instance is created. (instance creation)** **- The object/furniture is associated with the building it was created in. (association formed)** **- Building.gridWorld is updated with the new object. (attribute modification)** **- Building.minObjectCount is updated. (attribute modification)**

Contract CO6 — endBuildMode

Operation:	**endBuildMode()**
Cross References:	**Build Map**
Preconditions:	**- The player is in Building Mode.** **- The player has finished placing all objects/furniture for each building.**
Postconditions:	**- The Build Mode instance is deleted. (instance deletion)** **- A new Running Mode instance is created. (instance creation)** **- The instance ID attribute of the game is updated with the new running mode instance. (attribute modification)** **- All instances created in the Building Mode instance are associated with the newly created Running Mode instance. (association formed)**

Contract CO7 — spawnPlayer

Operation:	**spawnPlayer()**
Cross References:	**Run Game**
Preconditions:	**- Player is in Running Mode.**
Postconditions:	**- A new character and bag instance is created. (instance creation)** **- The player icon is associated with the map. (association formed)** **- The grid world attribute of the building is updated with the player's character icon. (attribute modification)**

Contract CO8 — playInBuilding

Operation:	**playInBuilding(building : Building)**
Cross References:	**Run Game**
Preconditions:	**- The player has completed Building Mode.** **- The player has entered Running Mode.**
Postconditions:	**- The character's existing attributes are saved and associated with the next building instance. (association formed)** **- The current bag instance is updated with the key removed, and associated with the next building instance with the other items remaining. (attribute modification, instance deletion)** **- The current building instance is deleted. (instance deletion)** **- The time attribute of the character is updated. (attribute modification)**

Contract CO9 — endRunningMode

Operation:	**endRunningMode()**
Cross References:	**Run Game**
Preconditions:	**- Player is in the running mode.** **- Player died or finished the game by collecting six keys and leave the final building.**
Postconditions:	**- The Running Mode instance is deleted, along with all of the instances contained in it. (instance deletion)** **- The instance ID for the game instance is updated with a result screen, showing the player their performance (game over screen or winning screen). (attribute modification)**

Contract CO10 — spawnAlien

Operation:	**spawnAlien()**
Cross References:	**Spawn Alien**
Preconditions:	**- The player is in Running Mode.** **- 10 seconds have passed since the last alien has spawned, or after the level begins.**
Postconditions:	**- An alien instance is created in the building the player is currently in. (instance creation)** **- The grid view attribute of the building is updated with the icon of the newly spawned alien. (attribute modification)**

Contract CO11 — deleteAliens

Operation:	**deleteAliens()**
Cross References:	**Spawn Alien**
Preconditions:	**- The player must be in Running Mode. - The player must have found the key to escape the current building they are in.**
Postconditions:	**- All alien instances in the building the player escapes from is deleted. (instance deletion)**

Contract CO12 — spawnPowerUp

Operation:	**spawnPowerUp()**
Cross References:	**Spawn/Despawn Power-up**
Preconditions:	**- The player is in Running Mode. - 12 seconds must have passed after the game begins, or a power-up spawns.**
Postconditions:	**- A new power-up instance is created. (instance creation) - The grid view attribute of the building is updated with the icon of the newly spawned power-up. (attribute modification).**

Contract CO13 — deletePowerUp

Operation:	**deletePowerUp()**
Cross References:	**Spawn/Despawn Power-up**
Preconditions:	**- The power-up must not have been collected in 6 seconds after it spawns.**
Postconditions:	**- The power-up instance is deleted. (instance deletion) - The grid view attribute of the building is updated with the power-up icon removed. (attribute modification)**

Contract CO14 — movePlayer

Operation:	**movePlayer(direction: int)**

Operation:	**movePlayer(direction: int)**
Cross References:	**Move Player**
Preconditions:	**- The player is in Running Mode.**
Postconditions:	**- The player's location attributes are updated in accordance with the specified direction. (attribute modification).** **- The grid view attribute of the building is updated with the new location of the player's icon. (attribute modification)**

Contract CO15 — collectItem

Operation:	**collectItem(itemID: String)**
Cross References:	**Collect Item**
Preconditions:	**- The player is in Running Mode.** **- The player must satisfy the distance requirements and the click type from the mouse, depending on the item collected.**
Postconditions:	**- The item and capacity instances of the bag are updated with the key or power-up instance added. (attribute modification)**

Contract CO16 — useKey

Operation:	**useKey()**
Cross References:	**Use Key**
Preconditions:	**- The player is in Running Mode.** **- The player must have the key stored in their bag.** **- The player must be close enough to a door.**
Postconditions:	**- Bag.items attribute is updated (attribute modification)** **- The key instance is deleted for the object/furniture and the Bag instance containing the key. (instance deletion)** **- Building.gridView attribute is updated with the unlocked door icon. (attribute modification)**

Contract CO17 — usePowerUp

Operation:	**usePowerUp(powerUpID : String)**
Cross References:	**Use Key**
Preconditions:	**- The player is in Running Mode.** **- The player must have the power-up obtained.**

	The player must have the power-up obtained.
Postconditions:	**- Various game and character attributes are updated based on the power-up used. (attribute modification)** **- Bag.items and Bag.capacity attributes are updated. (attribute modification)** **- The power-up instance is deleted, and the power-up is removed from the bag. (instance deletion)**

Contract CO18 — resumeGame

Operation:	**resumeGame()**
Cross References:	**Pause/Resume Game**
Preconditions:	**- The player is in the pause menu.** **- The player has pressed the escape or resume button.**
Postconditions:	**- RunningMode.modeID is updated with the main game session after resuming (attribute modification).**

Contract CO19 — pauseGame

Operation:	**pauseGame()**
Cross References:	**Pause/Resume Game**
Preconditions:	**- The player is in Running Mode.** **- The player has pressed the escape button.**
Postconditions:	**- RunningMode.modeID is updated with the pause screen. (attribute modification)**

Version History | Operation Contracts

Version	Date	Description	Author(s)
Draft v1	November 2, 2022	Created operation contracts for the use cases.	**Bartu Celasun, Eren Ceylan, Erim Satar, Hakan Çapuk**
Draft v2	November 3, 2022	Added more operation contracts	**Berfan Pirhan**
Revision	November 5, 2022	Updated some of the contracts	**Berfan Pirhan**
Revision	November 5, 2022	Added some new contracts.	**Bartu Celasun**

Revision	November 5, 2022	Updated some of the contracts	**Eren Ceylan**
Final	November 6, 2022	Finalized the contracts	**Berfan Pirhan**

Vision

❖ Introduction

This report is designated to document the requirements, goal, features and the progress of the team for this project.

References:

- COMP302 Lecture Slides taken from Koç University
- Applying UML and Patterns, Edition: 3E, Larman Pearson Education

❖ Positioning

❖ Business Opportunity

The gaming industry is one of the biggest sectors in software development. Within the gaming industry, 2D pixel retro games have been on the rise in the past years. As this game takes place in an academic institution, Koç University; this game will be intriguing and appealing for Koç University students. Because of this reason, there is a high demand opportunity for this version of an escape game, called “Escape From Koç”.

❖ Stakeholders & Users

❖ Stakeholders

- **Project Designers:** Want to create a 2D escape game based in an academic institution, while learning how to work in a team in a professional manner, adhering to agile development principles, and creating a sense of community within the school as members of the school.

❖ Users

- **Player:** Wants a 2D escape game that is playable, interactive, bug-free, easy-to-use, and entertaining.

Version History | Vision

Version	Date	Description	Author(s)
Draft v1	November 2, 2022	Initiated outline of vision	**Berfan Pirhan**
Draft v2	November 5, 2022	Most details for the vision added	**Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk**

Final	November 6, 2022	Finalized vision	**Berfan Pirhan**

Supplementary Specifications

❖ Introduction

This report is the repository of additional requirements of the game that are not covered in the use cases.

❖ Functionality

Security

Playing the game requires an email account and a unique username to register to the game. The player also uses their email address and the unique username they inputted with that email to log in to the game. Every player must log in to the game with a unique email and username.

❖ Reliability

The game needs to run smoothly, without bugs and system failures, and quickly re-collect itself if an error occurs.

❖ Usability

The game needs to run fast and in a smooth manner, and the visuals of the game needs to be clear and distinguishable from each other in order for the player to play the game comfortably.

Version History | Supplementary Specifications

Version	Date	Description	Author(s)
Final	November 5, 2022	Supplementary Specifications created.	**Bartu Celasun, Berfan Pirhan, Eren Ceylan, Erim Satar, Hakan Çapuk**

Glossary

Definitions

Term	Definition & Information	Aliases
Alien	Object/entity within the game that tries killing the player and prevents them from getting the key	

	from getting the key	
Key	Object that the player needs to collect to escape a building/room	
Power-up	Object that the player can use to ease the process of avoiding aliens and finding keys	
Building Mode	Game mode where the player designs the inside of buildings by placing objects	
Running Mode	Game mode where the player starts playing the game after building the map	
Player	Entity that is the main user for the game	
Character	Virtual representation of the player in the game, which the player controls	
The Gentlemen	Name of the team working on the project	
Git	Version control system	
GitHub	Service for software development and version control using Git	
Google Drive	Cloud storage system	Drive
Zoom Meetings	Video conferencing software for teams	Zoom
Integrated Development Environment	Software application that provides various mechanics for software development	IDE
Eclipse IDE	Integrated development environment mainly used with Java	Eclipse
Visual Studio Code	Code editing software	VSC
Java	An object-oriented and strongly-typed programming language	
Swing	Java UI library	

Bag	Players items storage	Inventory
Slack	Messaging program for developer teams	
Visual Paradigm	UML CASE tool for project management	
Draw.io	Open source graph drawing software	
Use Case	Description of tasks that will be performed in the program	
Use Case Narrative	Detailed description of a use case	
System Sequence Diagram	Visual representation of user-system interaction in use cases	SSD
Operation Contract	Detailed description of pre- and post- conditions of objects due to events	
Domain Model Diagram	Visual representation of objects and their connections	

Version History | Glossary

Version	Date	Description	Author(s)
Draft v1	November 1, 2022	Initiated glossary with some of the terms used	**Berfan Pirhan**
Draft v2	November 2, 2022	More terms added with their definitions and aliases	**Erim Satar**
Draft v3	November 3, 2022	More terms added with their definitions and aliases	**Berfan Pirhan**
Draft v4	November 3, 2022	More terms added with their definitions and aliases	**Eren Ceylan**
Final	November 6, 2022	Final terms added	**Berfan Pirhan**