peterjuskiewicz / **game-unity** Private

No description, website, or topics provided.

Edit

Manage topics

8 commits | **1** branch | **0** releases | **1** contributor

Branch: **master** | New pull request | Create new file | Upload files | Find File | Clone or download

peterjuskiewicz add concept — Latest commit `a2f15a7` 15 hours ago

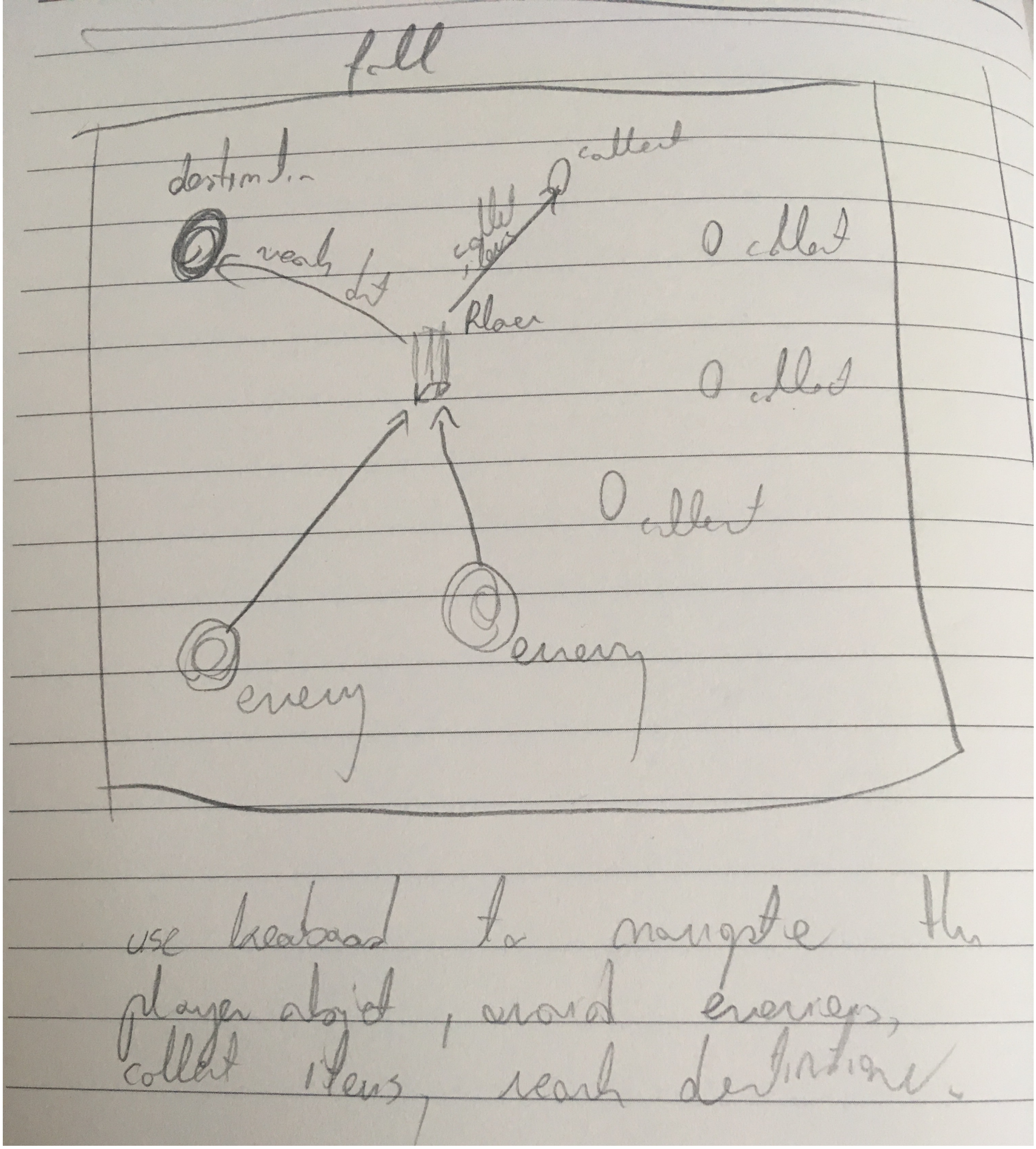
.DS_Store	add concept	15 hours ago
README.md	add concept	15 hours ago
concept.jpeg	add concept	15 hours ago
diagram1.png	add diagrams	2 days ago
diagram2.png	add diagrams	2 days ago
test.gif	add img	3 days ago

README.md

Design and Implementation

Concept

Victoria and Albert Museum hosted Video Game Exhibiton and that was the inspiration for me to build this game prototype. There were several games presented and all of them shared similar objectives: collecting items, killing or avoiding enemies and reaching destination. I have decided to build a game that will implement those 3 functionalities () using basic games objects. Below there are images showing the concept of the game.



Requirements

1. Player can navigate the player object by using the keyboard and mouse
2. The game is implemented in TPP
3. There are enemies objects that player has to avoid
4. There are collectables that player can collect
5. There is a destination that player has to reach in order do win the game
6. If player falls of the plane, he's moved to the spawn point.
7. If an enemy object collides with the player object, the game is over
8. If the player collects a collectable, the number of points increses by 1
9. If the player reaches the destination, he wins the game

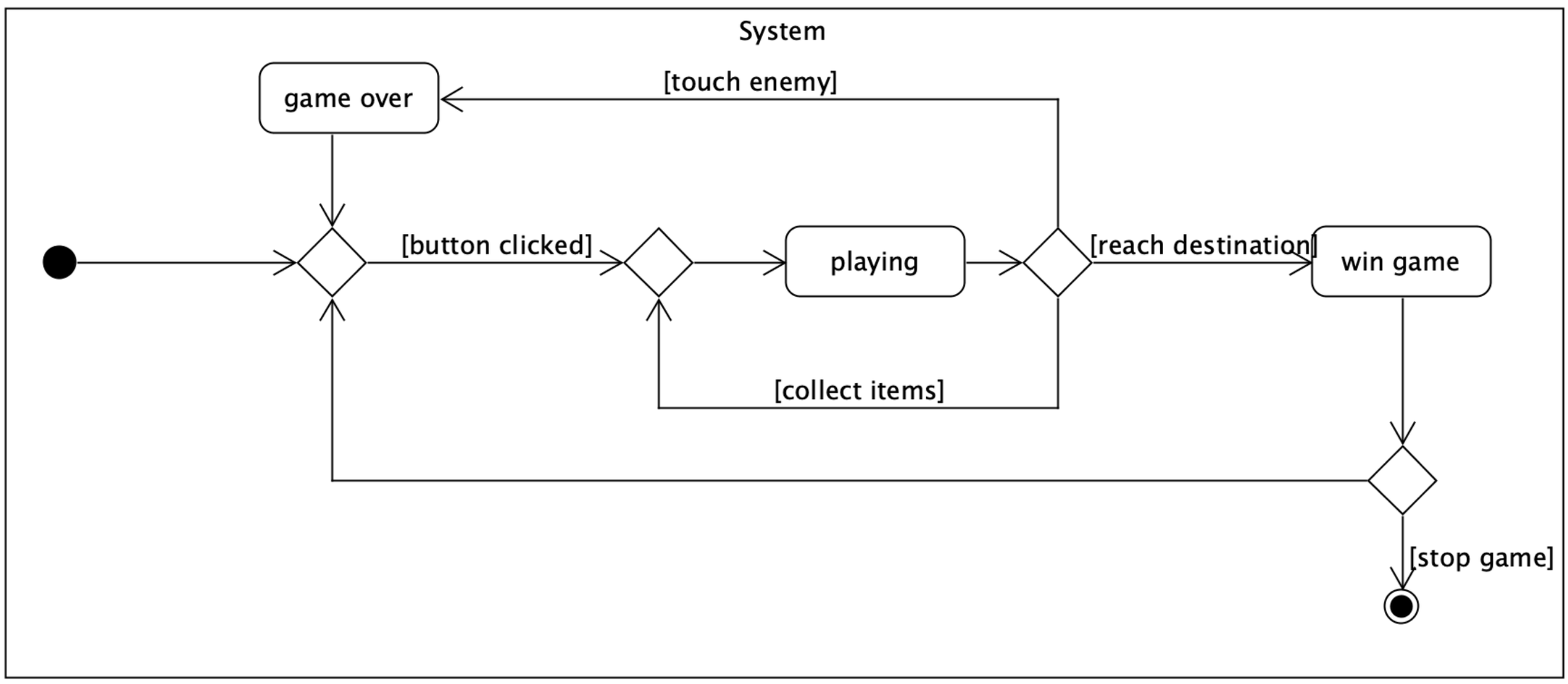
Assets requirements

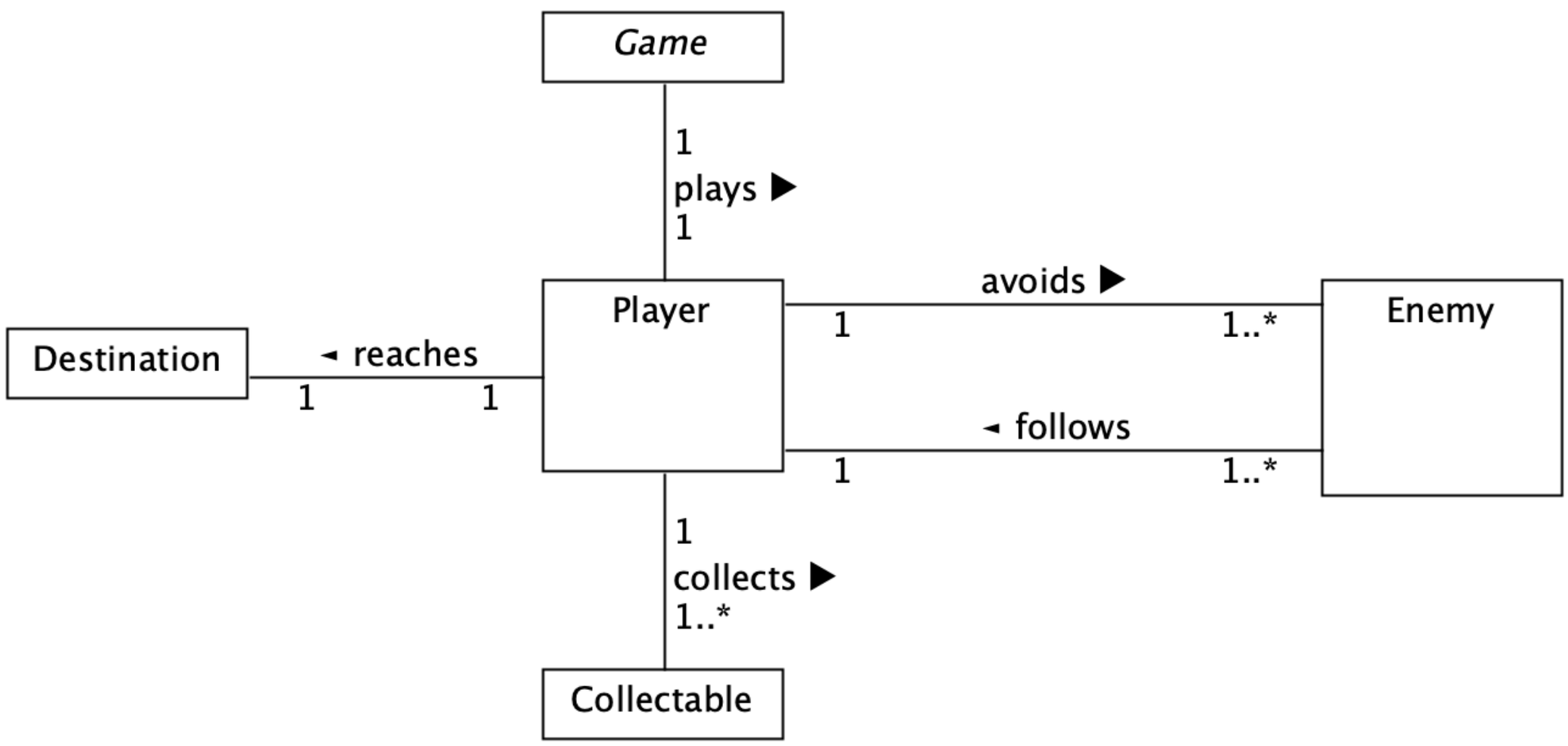
In this game projects only basic game objects are used: spheres, planes, cubes and capsules. Textures such as lava, grass, rocks and Skybox have been downloaded from the Asstets Store. All the basics objects can be easily replaced by assets. There are following objects in the game:

1. Player cube with white spot light attached and grass texture
2. Collectable capsules (5) with green halo effect
3. Enemy spheres (2) with red poin light attached and lava texture
4. Destination sphere (only collider) and the light pointing to the destination
5. Floor plane with grass texture, being walkable navigation area
6. Main camera following the player
7. Dimmed directional light

Analysis and design

The game will have a simple logic: player collecting items and trying to avoid enemies. The main goal of the player is to collect the highest number of collectable items and reach the destination where the game ends and the player wins. If the player is touched by an enemy the game is over. I also set up respawning in case the player object falls of the floor plane.





Implementation

1. Camera behaviour is implemented to simulate third person perspective. Camera is looking at the target object (Player) and follows the changing position of the object.
2. There is collision detection functionality - detect collistion with enemy, collectable and destination objects and call different functions (add points, win game, lose game) depending on which event has been triggered.
3. Player movement is implemented using turning and move methods.
4. Respawning of the player is implemented in the event of falling of the plane. There is an additional collider underneath the actual floor to detect the falling object.
5. There are UI elements displaying current state of the game (playing, win, game over) and number of collectables collected.

Object	Component	My Activity	Comments
Player (cube)	Transform	Set up the starting position	
	Mesh filter		
	Mesh renderer		
	Box collider		
	Script	Implemented player behaviour	Using online resources, official Unity docs, youtube videos and the unity handbook
	Rigidbody	Added, using gravity	
	Light	Added spot light	Flashlight effect
	Texture	Added grass texture	Downloaded from assets store
Enemy (sphere)	Transform	Set up the starting position	
	Mesh filter		
	Mesh renderer		
	Sphere collider		
	Nav mesh agent	Added humanoidto navigate the enemy	
	Script	Implemented enemy behaviour	Using online resources, official Unity docs, youtube videos and the unity handbook
	Light	Added red point light	Lava effect
	Texture	Lava pattern	Downloaded from assets store
Capsule collectable	Transform	Set up the starting position	
	Mesh filter		
	Mesh renderer		
	Capsule collider		
	Halo	Added to imitate halo effect	
	Texture	Grass texture	Downloaded form assets store
Destination	Transform	Set up the starting position	
	Mesh filter		

	Mesh renderer	Deactivated	
	Sphere collider		
	Spot light	Used to mark the destination	
Floor	Transform	Set up the starting position	
	Mesh filter		
	Mesh renderer		
	Mesh collider		
	Texture	Grass pattern	Downloaded from assets store
Plane	Transform	Set up the starting position	In size bigger than floor and the position is underneath the floor
	Mesh filter		
	Mesh renderer	Deactivated	
	Mesh collider		Player is respawned to the starting position if the collision is detected
Main Camera	Transform	Set up the starting position	
	Camera	Added skybox and background	Downloaded from assets store
	Audio listener		
	Script	Added camera behaviour	Follow and look at the player object
Directional light	Transform		
	Light	Change intensity and colour	
Canvas	Rect transform		
	Canvas		
	Script	Canvas scaler, Graphic raycaster	
	WinMessage	Added to display game state (playing, win, game over)	
	Points	Added to display the number of points	Incresed by 1 with every collectable capsule collected
StartGameButton	Rect Transform	Set up starting position	
	Canvas renderer		
	Image	Changed the colour	Deactivated when the button is clicked
	Script	Implemented onClick callback	Deactivates the image and sets the isPlaying to true

Evaluation

In the first attempt to build the project I was trying to implement the game using multiple assets from the Assets store that would include sounds, animation, etc. After spending several hours working with this Assets without implementing funcionalities in C# I have decided to build a prototype with basic game objects that can be easily replaced with actual assets and I focused on building behaviours. I was working with official unity documentation, and unity tutorials available on https://unity3d.com/learn and http://docs.unity3d.com. I have used pluralsight C# courses to expand my understanding and use of the language.

In PlayerBehaviour on Awake() the references to movement, playerRigidbody, buttonImage and startingPosition are initialized and the callback for the button.onClick is added. Collision detection is implemented in OnTriggerEnter(Collider other) and depending on the trigger different functions are performed.

There are 4 collision cases:

1. Player touches the enemy and the message Game over is displayed
2. Player collects the items and the score is increased
3. Player falls of the floor and is respawn to the starting position
4. Player reaches the destination and the message Win is displayed

I have used movement methods Move(h, v) and Turning() to be called in FixedUpdate() method.

In CameraBehaviour on Start() variables angle and offset are calculated and initialized. It implements follow the player fuctionality and on FixedUpdate setting the camera position with the offset relatively to the current position of the player. It also enables the player to use the mouse for rotation. The main problem with this behaviour was to ensure that the camera will always look at the player, hence I added LookAt() build in method.

In EnemyBehaviour on Awake() references to NavMeshAgent and Player are initialized and on Update() the destination of NavMeshAgent is set to the player current position such as the enemy can follow the player.

Personal statement

I can confirm that the work is my own. I have used official unity documentation, tutorials available on Unity website, unity game development handbook and pluralsight tutorials. I have downloaded several assets from Assets store.