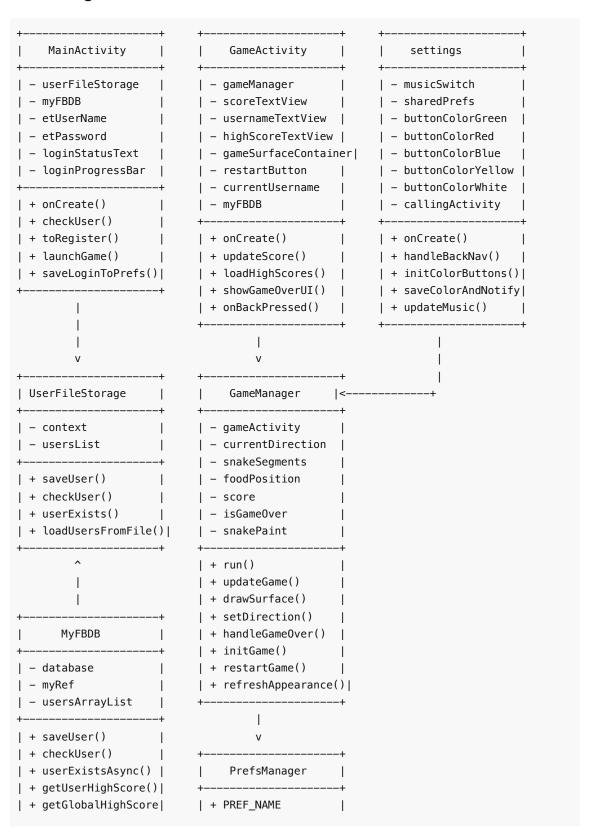
Snake Game UML Diagrams

Class Diagram



Activity Lifecycle Sequence Diagram

User Main	Activity Gam	eActivity	Settings	GameManager
1	1	I	1	I
> Launch				
	1			
	> onCreate()			
	Setup UI			
> Login		l	I	
1		l	I	
1	> checkUser()	•	1	
	Auth Proces	s	I	
1	1	I	I	I
1	> launchGame()	1	I
1	Intent	l	I	
		->	I	
	1	I	1	I
	1	> onCre	•	I
1	1	Setup	UI	I
1		l	I	
1		> loadH	ighScores()	
		l	I	
		> new G	ameManager()	
1	1		>	I
	1	I		<pre>> initGame() </pre>
1		l	1 1	
	1	I		> startGameLoop()
> Play Game	1	I	1 1	I
1	1	I		> run()
				Game Loop
			1 1	
> Settings				
		> setti	ngsLauncher.lau	nch()
			>	

	Setup UI
> Change Color	
	>
<pre>saveSnakeColorAndNotify() </pre>	
> Back	
	> handleBackNavigation()
	finish()
	<
	> onActivityResult()
	> gameManager.refreshAppearance()
	>
loadAndApplySnakeColor()	
> Game Over	
	> handleGameOver()
	<
	> showGameOverUI()
> Restart	i i i
i	> gameManager.restartGame()
i	>
i	> initGame()
 > Back	i i i i
i	> onBackPressed()
	Show Dialog
 > Confirm Exit	
Company External	
	-
ı	

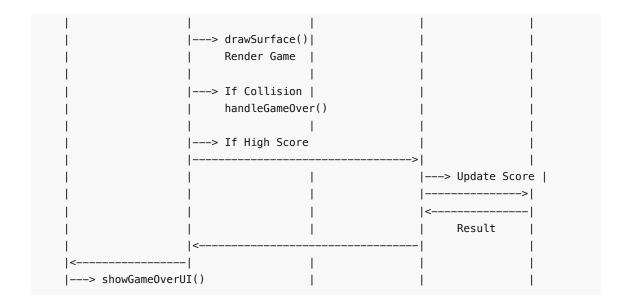
Authentication Sequence Diagram

User	MainActivity	UserFileStorage	MyFBDB	Firebase
1	1		1	
> I	nput Credentials	1	1	1
			1	
	> chec	ckUser()	l	1
		>	1	
1		> Check	Local Storage	e
			1	
	<		1	

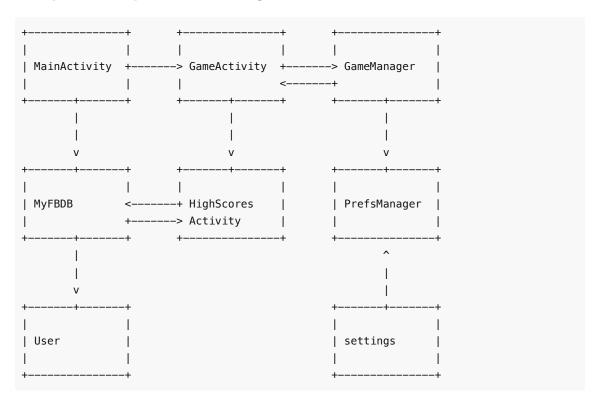
	Result		1
>	I If not found, userExistsAsync()	
	> 	 > Query User	
	 	> <	· ·
	 	Result 	
	Result	 	
>	<pre>If exists, checkUserAsync()></pre>	 	I I
	1	> Verify Pwd	
	İ	 < Result	.
<	I	Nesucc	
	Result	 	I I
	If valid, saveUser() >	 	
	Save to local	 	
>	launchGame()		1

Game Logic Sequence Diagram

GameActivity	GameManager	PrefsManager	MyFBDB	Firebase
1	1	1	1	
> initGam	ne()	1	1	
	>	I	I	1
I	> getSna	akeColor()		I
I		>	1	I
I	<		I	
ļ	Color	ļ	l	l
			. 1	I
l l	> Setup	Game State	1.	l .
		ļ		
> Game Lo	oop	ļ	l	l
		-Cama()	ļ	
I	> update Move 9		l I	l I
I	·	Collisions	l I	l I
l I	Check		l I	l I
I I	l clicck	1 00u	l I	l I
	> If Fo	nd Faten	i i	i I
	•	ment Score		
<		i	i I	i I
> updateS	score()	i		



Component Dependencies Diagram



Notes on UML Diagram Usage

The UML diagrams above provide a comprehensive view of the Snake Game application's architecture:

- 1. Class Diagram: Shows the main classes, their attributes, methods, and relationships.
- 2. **Activity Lifecycle Sequence Diagram**: Illustrates the flow of activity transitions and how they interact during the app's lifecycle, from launch to game over scenarios.

- 3. **Authentication Sequence Diagram**: Details the authentication process, showing how credentials are verified against local storage and Firebase.
- 4. **Game Logic Sequence Diagram**: Focuses on the main game loop, collision detection, scoring, and interaction with Firebase for high score updates.
- 5. **Component Dependencies Diagram**: Provides a high-level overview of how different components in the application depend on each other.

These diagrams can be used for:

- Onboarding new developers to understand the application structure
- Planning future enhancements by understanding current component relationships
- Identifying potential areas for refactoring or optimization
- Documenting the implemented architecture for future reference