Mobile Game: Heroes of the Rift

by

Jared Hilgefort, Jacob Parmley, Bunty Ranu, Greg Szczublewski

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Jared Hilgefort, Jacob Parmley, Bunty Ranu, Greg Szczublewski 4/23/2014
Jared Hilgefort, Jacob Parmley, Bunty Ranu, Greg Szczublewski Date

Russell E McMahon 4/17/2014
Russel McMahon, Faculty Advisor Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2014

ACKNOWLEDGMENTS

We would like to recognize individuals who have contributed to the success of *Heroes of the Rift*.

Professor Russ McMahon, our technical advisor.

Jim Scott and Professor Patrick Kumpf, our project advisors.

Michael Clark, Jr. and Michael Kirk, professionals working in the world of mobile applications who also have strong interests in mobile gaming and mobile game development.

Caitlin McGinn, our graphic designer.

Randall Rigdon, our audio designer.

Emily Routt, our concept artist.

Jacob Moore, our server host.

All of our professors at the University of Cincinnati, who imparted the knowledge on us to make this project possible.

All of our testers, who helped us to squash bugs early and often.

ABSTRACT

Heroes of the Rift is a free to play social action game developed for casual mobile gamers. Players will enlist the help of Facebook friends to venture into the Rift and harvest Buntonium. In the Rift players will encounter fast-paced combat, unlock weapon upgrades and take full command of the battlefield using intuitive touch controls. Game play is fast-paced due to our Thrusting Auto Lock-ON (TALON) combat system. Using TALON, players are able to dash around the level with the swipe of a finger, locking-on to and engaging the nearest enemy. Enemy clusters are randomly generated, making every mission feel unique. Server synchronization will enable player account information to be seamlessly shared across all of a user's devices – creating a truly mobile experience. Play with your character, your friends, on any device, at any time.

TABLE OF CONTENTS

Section **Page Number**

PROBLEM NEED

Problem Description

The mobile game market has grown exponentially as the number of phones and tablets have proliferated, making every owner a potential gamer. The most popular download on mobile platforms is currently games[1].

The nature of the mobile platform promotes gaming socially through involving friends and sharing a player's experience on social media. This platform shift has birthed a new type of player: the casual social gamer. The casual social gamer is an individual who plays simple games in which the experience is shared amongst friends.

Through observation, our team has noticed that there is a lack of fast-paced action games in the market which appeal to the casual social gamer. Games which are simple do not tend to be fast-paced; conversely, games which are fast-paced do not tend to be simple. This is because action games tend to be complex and naturally appeal to traditional gamers who do prefer console gaming to the pick-up-and-play nature of mobile devices.

Technical Areas

This project encompasses all areas of our academic discipline, but focuses primarily on two: software development and database.

Software development

This project involved creating a mobile game using the Unity3d game engine which is based upon C# and Unity script, a class-based JavaScript derivative. It also required the use of PHP scripting to communicate with a centralized database.

Database

This project required designing two databases which had to communicate with each other. The first is a MySQL database which is located on a server and serves as the central repository. The second database is a SQLite database hosted on client devices which holds a cache of data allowing players to play offline. We use prepared statements to make safer network calls to the MySQL database from mobile clients.

Other Technical Areas

In addition to the software and database components, this project will requires digital media design and intelligent use of network operations from a software development perspective. One of the standout characteristics of a successful game is its artistic design, which is why it was critical that we focus attention on modeling and designing our audio and visual components. With optimizing the user experience in mind, we designed data synchronization in a way that creates low impact to the user.

PROJECT DEFINITION

Heroes of the Rift involved designing and developing a socially interactive, 3rd person isometric shooter game for mobile platforms, specifically Android 4.0 and greater. This project existed within a full software development lifecycle (SDLC) project using SCRUM agile methodologies. We developed game play using the Unity3d game engine. We will manage information using PHP scripts and a MySQL database on a standalone server.

The game is a fast-paced isometric shooter. Players enlist the help of friends via social networks to complete missions. Game play is a passive cooperative experience in which the stats of your friends' characters are passively applied to your own; they do not have any direct influence on game play. Our game incorporates the Thrusting Auto Lock-ON (TALON) system to create fast-paced game play. Contributing to this fast-paced game play is our "stream vs. cluster" enemy spawn algorithm. The "cluster" spawns clusters of enemies across the map which must be eliminated to complete the mission. The "stream" sends a constant wave of enemies toward the player originating from enemy "clusters."

Detailed Game Play Description

At its core, *Heroes of the Rift* is a fixed 3rd person isometric shooter. Fixed 3rd person isometric describes the camera angle during play. 3rd person means players are watching their avatar from an exterior perspective. Fixed isometric refers to a static camera at an angle which is approximately 45 degrees relative to the surface of the game world. Shooters are games in which the primary mode of game play is through projectile combat. The game includes elements of role playing games or RPGs. RPGs are games that allow players to increase their skills through playing the game and earnign experience.

The game incorporates a passive social experience to enhance game play. Players can select a limited number of friends to join their squad. The avatars of a player's squad mates join the player in-game and provide a boost to the player's health and thruster capacity.

The game loads to the menu screens. From the menu screens, users are able to:

1. manage their account, squad, and thrusters,
2. view and compare stats from game play,

3. select missions, and
4. launch a game.

These screens are all seamlessly and intuitively navigable from each other.

Once a game has been launched, the player's objectives are to destroy all of the Hulk enemy types and to collect resources. The player controls directional movement using a touch joystick, fire projectiles by tapping or holding the screen, and dash/thrust directionally by swiping across the screen.

Combat is fast paced due to spawning large numbers of weak enemies across the map. Players move across the map to engage all of these enemies.

Through combat and game play, players level up their avatars.

Within the RPG system there are be three classes of players, each with a special skill. The special skills can be activated and last for a limited time. The skills and classes are:

1. *Assault* – attack focused class
    a. The Assault's skill allows targeting of the two nearest enemies
2. *Guard* – defense focused class
    a. The Guard's skill allows an electromagnetic pulse to disable enemies
3. *Scout* – movement focused class
    a. The Scout's skill enables a thrust to be used as an attack

How it Solves the Problem Need

Heroes of the Rift is a fast-paced action game that incorporates social experiences through the implementation of passive cooperative play and simple RPG elements. This type of game play fills the void which we have identified as a lack of fast-paced action games designed for the casual mobile gamer.

Scope and Resources

User Profile

Potential users for *Heroes of the Rift* include Android users with Android 4.0 or greater. The game was designed with a focus on interacting socially with friends and other players. The genre of the game is an action shooter which attracts traditional gamers. Through simple game play, sharing stats, and earning rewards for friends the social appeal reaches beyond traditional gamers. Through game balancing we and system fine tuning, we keep all players engaged.

Our application was built using the established fundamentals of software interface design. The game interface communicates game play functionalities clearly. Users familiar with mobile touch interfaces will recognize similarities and standards which we aimed to abide by. Navigating menus is as simple as tapping or swiping the touch screen.

Game play involves a series of touch, swipe, and shaking motions. Users familiar with mobile gaming have likely been exposed to something comparable.

Any user that is comfortable using mobile devices to play games should feel comfortable playing *Heroes of the Rift*; similar experiences can be found in mobile games, regardless of genre. Users that are comfortable using mobile touch screen devices (not necessarily games), will notice familiarity during use. During game start-up, users will be given a tutorial that describes core mechanics and social interaction. Once the tutorial is completed, users will be equipped with the skills necessary to complete additional game play tasks.

Given the nature of mobile gaming, we have designed *Heroes of the Rift* to be played in short intervals. These intervals range from one to ten minutes, allowing a "pick-up-and-play" type of experience. Players can quickly complete a mission, go on to doing something else, and come back to play another mission.

Project Cost

The estimated budget of this project is approximately $7000. Our estimated hardware costs are included as if we did not own the development laptops and mobile devices. The server cost reflects a monthly fee we would pay had we not been allowed free database hosting courtesy of Jacob Moore. Our business costs cover the incorporation of a limited liability corporation, filing for our employer identification number, and various printing costs. We counted any free or purchased software or licenses as part of our software budget. Unity3d has a free version which has an open license allowing us to sell our game under that license. LAMP server stack is free open-source software that requires us to do all of our own server and database management.

Hardware Budget

Product	License Type	Unit Cost	Number Purchase	License Term (months)	Total
Laptop		$1,000.00	4	-	$4,000.00
Server		$ 50.00	1	12	$ 600.00
Mobile Device		$ 350.00	4	-	$1,400.00
				Hardware Total	**$6,000.00**

Business Costs

Product / Service	License Type	Unit Cost	Number Purchase	License Term (months)	Total
LLC incorporation	Filing fee	$ 125.00	1	-	$ 125.00
EIN application	Filing fee	$ 60.00	1	-	$ 60.00
Filing Service	filing service	$ 149.95	1	-	$ 149.95
				Business Costs Total	**$ 334.95**

Software Budget

Product / Service	License Type	Unit Cost	Number Purchase	License Term (months)	Total
Unity	Free	$ -	4	-	$ -
Autodesk 3ds Max	license	$ 200.00	1	2	$ 400.00
Android SDK	Open-source	$ -	4	-	$ -
NeatPlug Universal Social plugin	license	$ 80.00	1	-	$ 80.00
MySql	Open-source	$ -	1	-	$ -
Apphance	free license	$ -	4	-	$ -

PHP	free license	$ -	4	-	$ -
Apache	free license	$ -	4	-	$ -
Unity Assets - free	free license	$ -	4	-	$ -
Unity Assets - paid	license	$ 30.00	2	-	$ 60.00
Android Key	license	$ 25.00	1	-	$ 25.00
Apple Dev Key	license	$ 99.00	1	12	$ 99.00
				Software Total	**$ 664.00**

Grand Total of Costs **$ 6,998.95**

Figure 1. Estimated Budgets for Hardware, Business, and Software

Timeline

Figure 2 below shows a high-level timeline for the duration of this project. The core development will occur from December through January. February will be dedicated to integration of assets and ensuring proper synchronization with the database. At the end of each month the will be a week of testing among volunteer users. In March, we will do heavy User Acceptance Testing (UAT). April will be dedicated to preparing for the 2014 IT Expo.

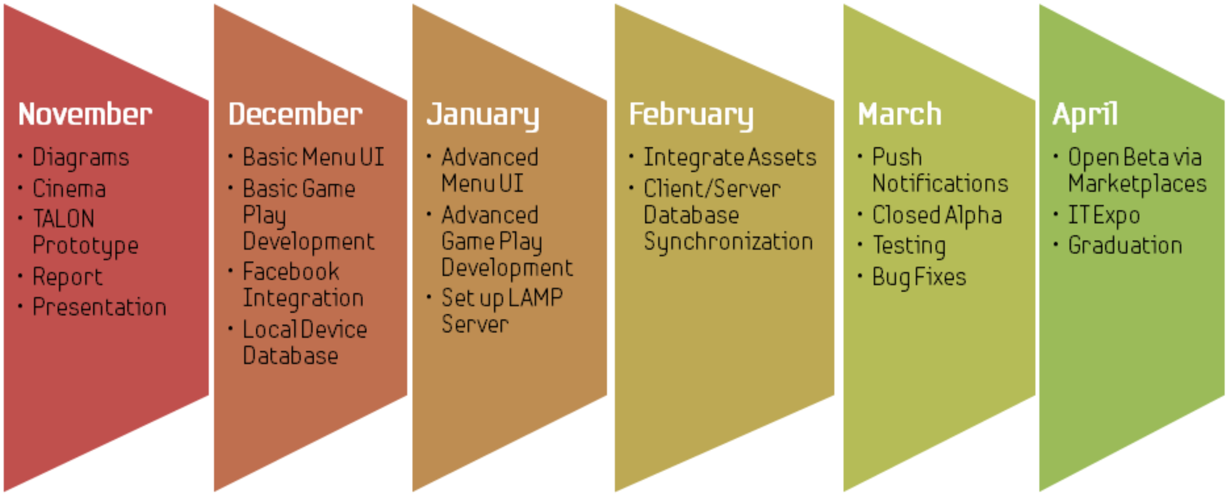


Figure 2. Project Timeline

Hardware and Software Resources

Creating *Heroes of the Rift* requires an assortment of tools to complete tasks such as graphic design, audio design, game play development, mobile development, and database management. The tools can be split into two logical categories: hardware and software.

Hardware

Development computers - This project required all four members to have access to development laptops so that we can easily meet and work together or work separately on our own schedule.

Mobile devices - This project required each developer to have an Android device running Android 4.0 or higher for testing. Among our group we already own a Nexus S 4G, Samsung Galaxy S3, two Samsung Galaxy S4s, a Samsung Galaxy Nexus, Samsung Galaxy Tab 3, and a first generation Nexus 7 tablet.

Server - This project required a web application server. We have chosen to do a LAMP implementation.

Software

3D Studio Max – 3d Studio Max was used to create meshes, character models and map materials representing graphical objects within the game.

Adobe Creative Suite – The creative suite was used to create graphical elements within the game. Photoshop was used for conceptual art, painting models and creating graphical components. Illustrator was used for heads up display elements and the custom *Heroes of the Rift* logo font.

Apache – Apache is a web service which runs on a Linux platform.

GitHub Enterprise – GitHub is an online code repository which enables easy sharing and tracking of code.

Google Drive – Google Drive is a shared online document repository which we leverage for managing the day-to-day project details.

JSON – JSON is a light-weight data-interchange format based on JavaScript which is easy for applications to parse.

Linux – Linux is a flexible Operating System which can run applications by implementing the LAMP (Linux Apache MySQL PHP) stack of technologies.

MySQL - MySQL is an open-source database which we use as the central repository for our game.

Neatplug Universal SNS plugin for Unity - We integrated Facebook functionality into the game to promote its social nature. Facebook features include: user profile name, user profile picture friend invites, “Like” button, and a “Share” button to post to your feed or a friend’s wall.

PHP – Communication between the client devices and the database required the use of a web service running on the server alongside the MySQL database. To address this need we used PHP scripting.

SQLite – SQLite is is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

Unity3d – Unity3d is a free 3-D game engine that provided us a platform on which to develop the game play without needing to develop the underlying functionality.

Mobile Application Testing

Initially our group looked to implement both unit testing and user acceptance test cases. Due to the learning curve we experienced with Unity integrated development environment, the group decided the unit testing task added additional development overhead that may have put our schedule off by several months. We do recognized that the unit tests would have allowed fine grain testing of component pieces and built in regression testing but the task would be time consuming for those of us that are not familiar with best practices and mocking in unit testing. Instead we opted to leverage a test case tree document to validate application features and functionality. This approach is only available to the developers and a select set of individuals with testing procedure knowledge. One recognized downfall to this approach is whenever component functionality drastically changes; the parent and child acceptance tests related to the component we require retesting. Thus far, this downfall has not limited our development or testing timetables.

Test Case Tree

A test case tree assists in documenting the steps for testing a unit of functionality via a descriptive acceptance test. The parent-child relationships in the test case tree document the repetitive steps to ensure coverage to each child node in the test case tree. The test case tree acts as a map for efficiently writing the user acceptance tests that the tree supports. The test case trees also documents which acceptance test cases have been written, the date of passing or failing, and that data is conveyed easily by a color coding scheme to which our group must adhere.

Apphance Integration

We chose to leverage Apphance uTest Unity integration to record device state and debug information of testers. Apphance automatically sends crash reports of users to its website which allows developers to monitor bad code. Apphance also allows users to record bugs manually in Quality Assuarance (QA) mode or to provide feedback in Production mode.

The integration within Unity was very simple. We only needed to include the Apphance SDK, attach an Apphance script to a game object within every scene, and add our application keys to the script.

Release Cycle

As an agile project we chose to release portions of the game iteratively to allow users to test our product as we continued to develop it. This allowed us to both find bugs in the code and to get feedback and rapidly identify and fix any issues. Our goal was to do one release every month and allow our testers to test that build at their own leisure while we test code currently in development.

Risks

There are several risks involved in completing the project. Primary risks are examined below.

Time

We used agile scrum development to manage time on individual tasks. Our sprint stories ensured key components for game play and evaluation were completed and resistant to scope creep based on sprint tasks. We have assigned various roles to ensure focus on each domain. Lastly, we enlisted the help of experts to assist in managing unexpected concerns or problems.

Scope

Maintaining scope is a large risk when developing games. Starting with a huge idea with near limitless game play opportunities is often a starting point. Then, through evaluation of assets, time, tools and knowledge the scope is refined and its core established. To solidify the scope, we have:

- chosen powerful tools (Unity3d and 3DS Max)
- defined core game play requirements
- prioritized functionality and design
- scrummed out the project into sprint tasks.

Cost

Cost is a risk when considering any development project. Luckily, we have found free tools (Unity3d, LAMP server stack) with distributable licenses. We have the means to host the application on a LAMP server. We have explored opportunities for profit (out-of-scope micro-transactions). We have taken special care to ensure the cost has been estimated within our budget.

Communication

Without proper communication during a software development lifecycle, you risk reworking sections, muddling project vision and missing proposed deadlines. To mitigate these risks we have assigned roles and are using scrum development. We have a weekly meeting rhythm. We use various channels of communication, such as text and email, to ensure everyone is on pace. To manage code and communicate document changes, we have created a document repository on Google drive. We also used a code repository on GitHub to allow us to work on the project simultaneously.

Requirements

We have documented nearly all project requirements. At the end of every sprint we evaluated how well we have met proposed requirements.

Inexperience

Within our group, we have a strong background in development, game design, project management, and testing. We feel that we lack experience in agile/scrum methodologies and releasing products to the market; therefore we have enlisted the help of Scrum Masters as experts advising our group who have experience managing and releasing mobile applications for Kroger.

Design Protocol

Below are design documents describing technical elements of *Heroes of the Rift*. Included are:

- User Interface Design
- Social Integration
- Database Design
- Audio and Graphical Design

These areas define the elements which development tasks will based upon.

User Interface Design

Use Case Diagram

The actors in *Heroes of the Rift* are all users of the game, which we will refer to as players. Players have two primary groups of actions: pregame and in-game. Figure 3 below shows all of the pregame actions a player can accomplish.

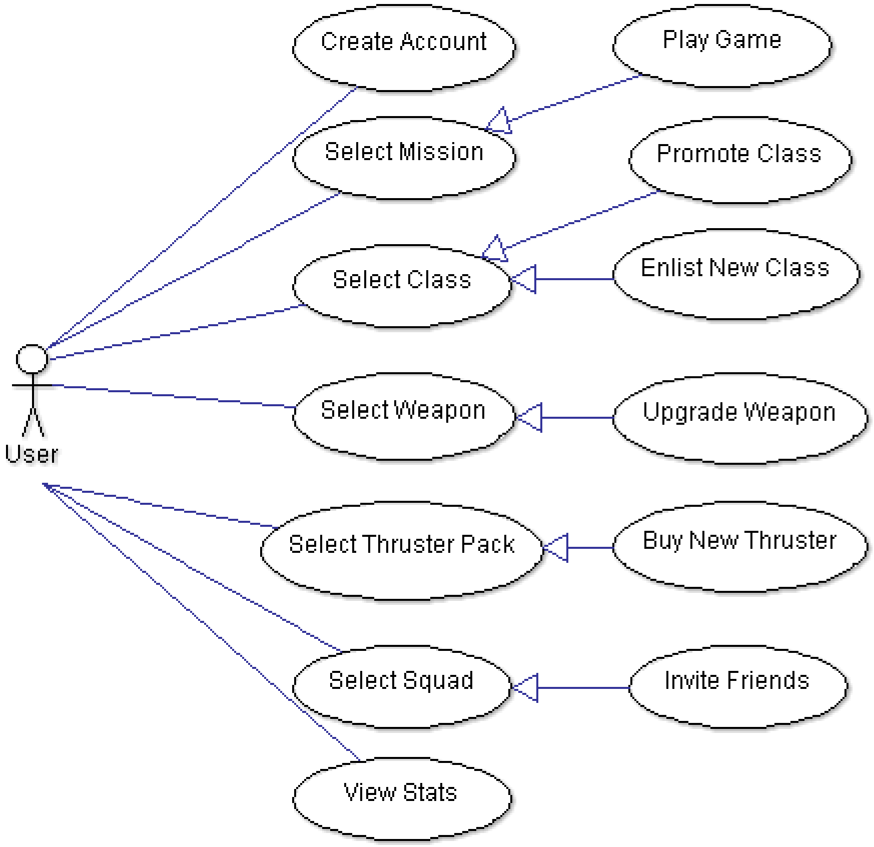


Figure 3. Use Case Diagram

Pregame actions include anything done in the menus, such as: creating an account, selecting a mission, selecting a class, selecting or buying a thruster pack, selecting or growing their squads, and viewing stats.

In-game actions include moving, shooting, thrusting, and using special abilities.

User Interface Model Diagram

There are two distinct screens within the game: the pregame menu and the game. The pregame menu, shown in Figure 4 below, will consist of five segments:

1. The first segment is the menu interface which is a dynamic frame which displays content based what the player is currently trying to do. There are five menus displayed in the frame: Settings, Mission Select, the Armory, Squad Select, and Stats.

2. The second segment is the navigation controller for the menu interface; it allows the player to scroll through the various menu screens.

3. The third segment is the user panel; it allows the player to log into Facebook, quickly change their class and invite friends to download the game.

4. The fourth segment is the squad screen; it allows the player the view his or her experience, the effect the squad has on stats, and overall squad rating.

5. The fifth segment is the squad viewer; it allows the player to scroll through squad members to view each members affect on the player's stats.

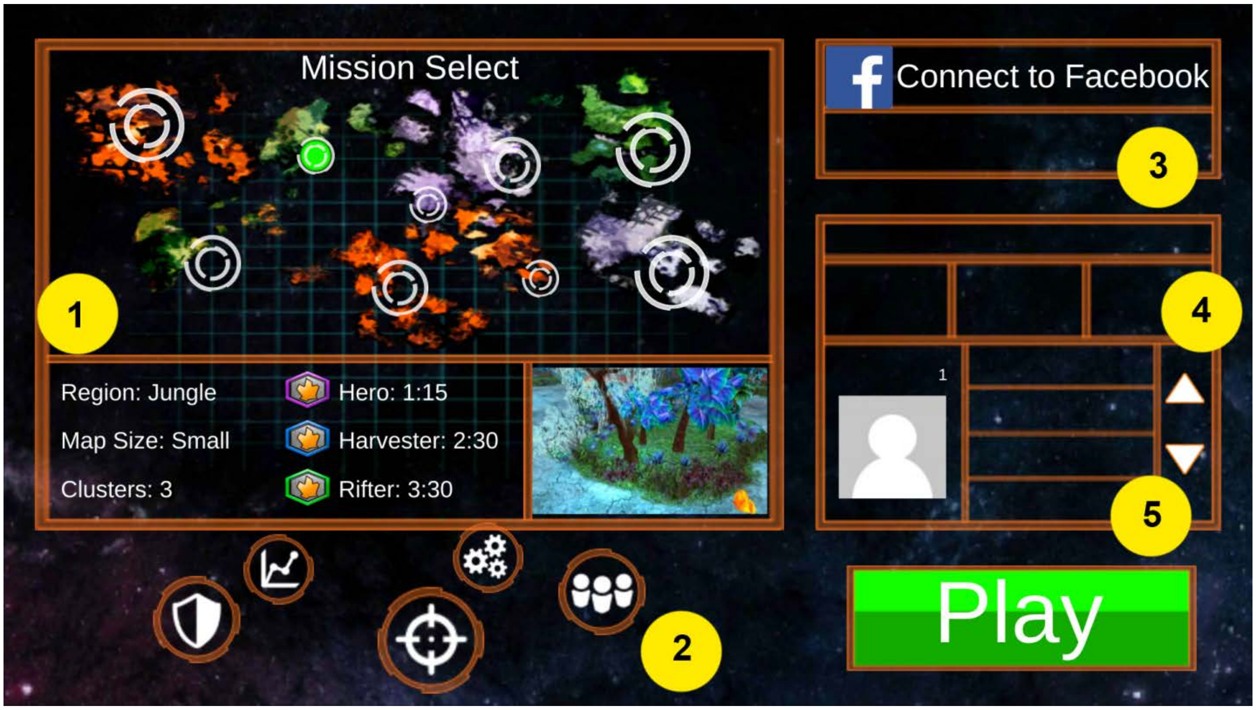


Figure 4. UI Diagram - Menu

The game screen, shown in Figure 5 below, demonstrates the look of the game to the player. The game screen consists of nine segments:

1. The joystick allows players to control directional movement.

2. The pause button allows the player to enter an in-game menu and halts action within the game.

3. The player and squad are the graphical representation of the player. This is the player controller object within the game.

4. Enemies will come in multiple types with various properties. Destroying the enemies earns the player points and add to the special meter bar.

5. The world consists of destructible environment which give the player additional points. Environment can only be destroyed by tapping the game play object.

6. The score shows the amount of points the player has earned.

7. The completion progress tracks the number of Hulk enemy types destroyed and the number currently remaining.

In addition to the controls specified above, players may tap anywhere on the screen (where another control doesn't exist) to fire his or her weapon and may swipe to dash in the direction of the swipe gesture.

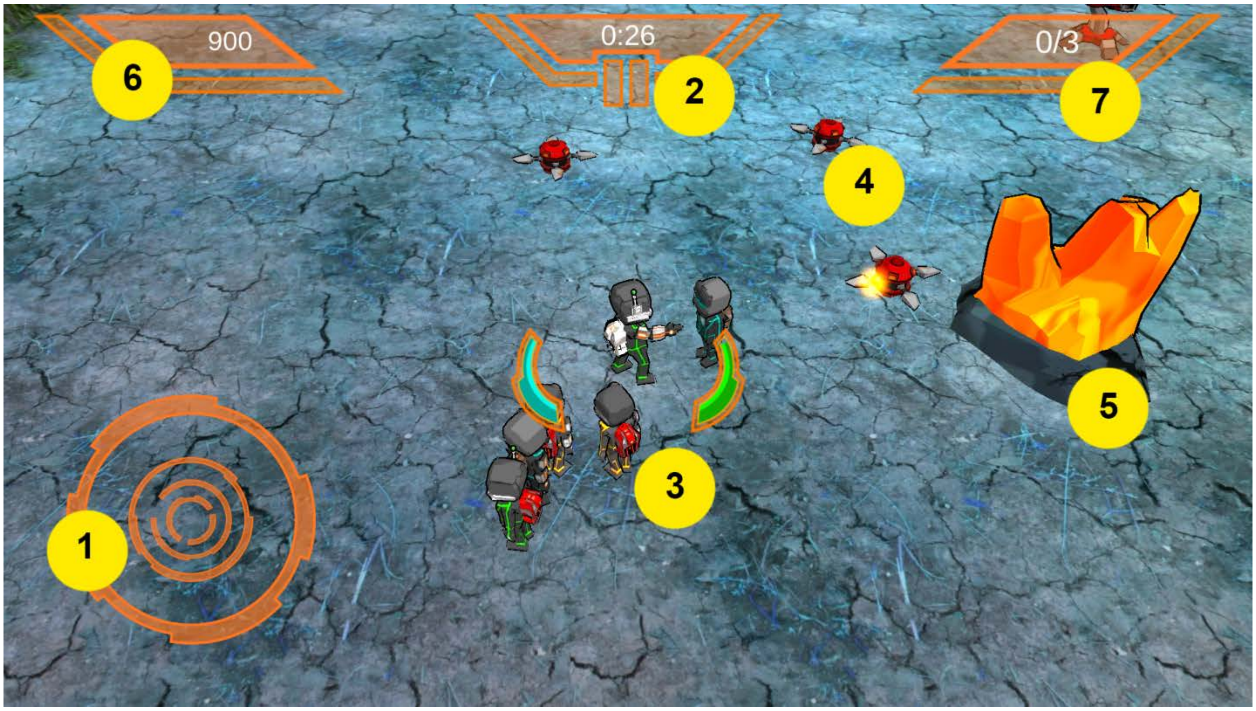


Figure 5. UI Diagram - Game

UI Flow Diagram

The UI flow diagram located below in Figure 6 demonstrates what data is to be shown on each screen of the application and how the user can navigate from one screen to another. The diagram also describes how the user can interact with the screen via touch controls.

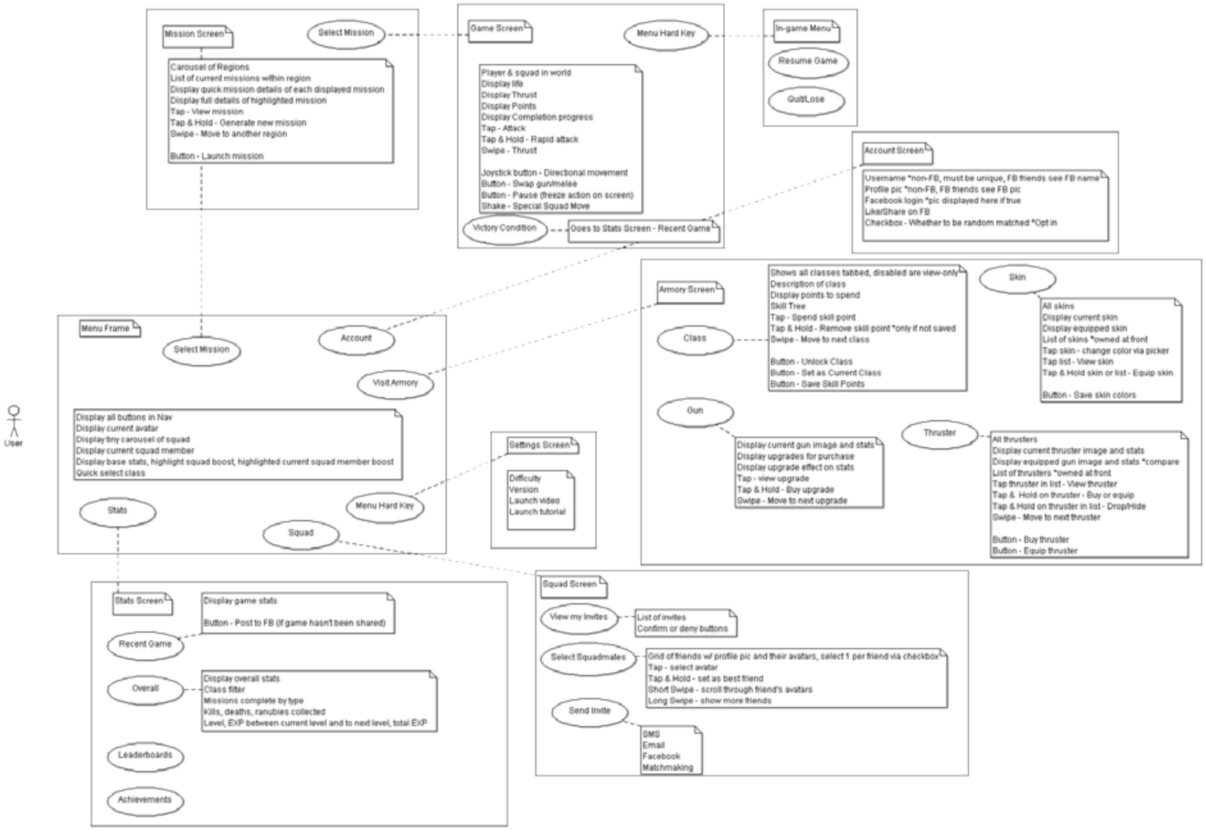

Mission Screen
Select Mission
Carousel of Regions
List of current missions within region
Display quick mission details of each displayed mission
Display full details of highlighted mission
Tap - View mission
Tap & Hold - Generate new mission
Swipe - Move to another region
Button - Launch mission
Game Screen
Menu Hard Key
Player & squad in world
Display life
Display Thrust
Display Points
Display Completion progress
Tap - Attack
Tap & Hold - Rapid attack
Swipe - Thrust
Joystick button - Directional movement
Button - Swap gun/melee
Button - Pause (freeze action on screen)
Shake - Special Squad Move
Victory Condition
Goes to Stats Screen - Recent Game
In-game Menu
Resume Game
Quit/Lose
Account Screen
Username *non-FB, must be unique, FB friends see FB name
Profile pic *non-FB, FB friends see FB pic
Facebook login *pic displayed here if true
Like/Share on FB
Checkbox - Whether to be random matched *Opt in
Armory Screen
Shows all classes tabbed, disabled are view-only
Description of class
Display points to spend
Skill Tree
Tap - Spend skill point
Tap & Hold - Remove skill point *only if not saved
Swipe - Move to next class
Button - Unlock Class
Button - Set as Current Class
Button - Save Skill Points
Class
Skin
All skins
Display current skin
Display equipped skin
List of skins *owned at front
Tap skin - change color via picker
Tap list - View skin
Tap & Hold skin or list - Equip skin
Button - Save skin colors
Gun
Display current gun image and stats
Display upgrades for purchase
Display upgrade effect on stats
Tap - view upgrade
Tap & Hold - Buy upgrade
Swipe - Move to next upgrade
Thruster
All thrusters
Display current thruster image and stats
Display equipped gun image and stats *compare
List of thrusters *owned at front
Tap thruster in list - View thruster
Tap & Hold on thruster - Buy or equip
Tap & Hold on thruster in list - Drop/Hide
Swipe - Move to next thruster
Button - Buy thruster
Button - Equip thruster
Menu Frame
Select Mission
Account
Visit Armory
User
Display all buttons in Nav
Display current avatar
Display tiny carousel of squad
Display current squad member
Display base stats, highlight squad boost, highlighted current squad member boost
Quick select class
Stats
Menu Hard Key
Squad
Settings Screen
Difficulty
Version
Launch video
Launch tutorial
Stats Screen
Display game stats
Button - Post to FB (if game hasn't been shared)
Recent Game
Overall
Display overall stats
Class filter
Missions complete by type
Kills, deaths, ranubies collected
Level, EXP between current level and to next level, total EXP
Leaderboards
Achievements
Squad Screen
View my Invites
List of invites
Confirm or deny buttons
Select Squadmates
Grid of friends w/ profile pic and their avatars, select 1 per friend via checkbox
Tap - select avatar
Tap & Hold - set as best friend
Short Swipe - scroll through friend's avatars
Long Swipe - show more friends
Send Invite
SMS
Email
Facebook
Matchmaking

Figure 6. UI Flow Diagram

Social Integration

Heroes of the Rift incorporates social elements as a core of the overall experience. The concept of creating a squad of friends that boost stats and assist in game play is unique to the project. The integration of this social aspect drives the game closer to our target audience, while innovating on established game play principles. Players are able to distinguish friends as squad mates based on the appearance and style of their avatar. This visual representation of friends within a squad strengthens the bond between the player and squad.

Facebook

To incorporate the social nature we leveraged Facebook via Neatplug SDK. The Facebook platform is well established in mobile games, allowing players to connect with existing friends rather than requiring standalone accounts and requests within our game. A player's Facebook account is his/her primary account in *Heroes of the Rift*; we use the player's name and profile picture as they appear on Facebook, and automatically load any friends who have also downloaded the game as potential squad mates.

Database Design

To optimize the experience for mobile gamers, we incorporated a centralized server to allow synchronization across multiple devices (phones and tablets) and sharing of information amongst friends.

Client/Server Architecture

Data is pushed and pulled from a MySQL database on a centralized server using PHP, as shown in Figure 7 below, to enable players to play across multiple devices as well as to enable synchronization of data with friends. MySQL transactions are managed through a list of stored procedures which are called using PHP via HTTPS POST methods sending and retrieving JSON arrays.

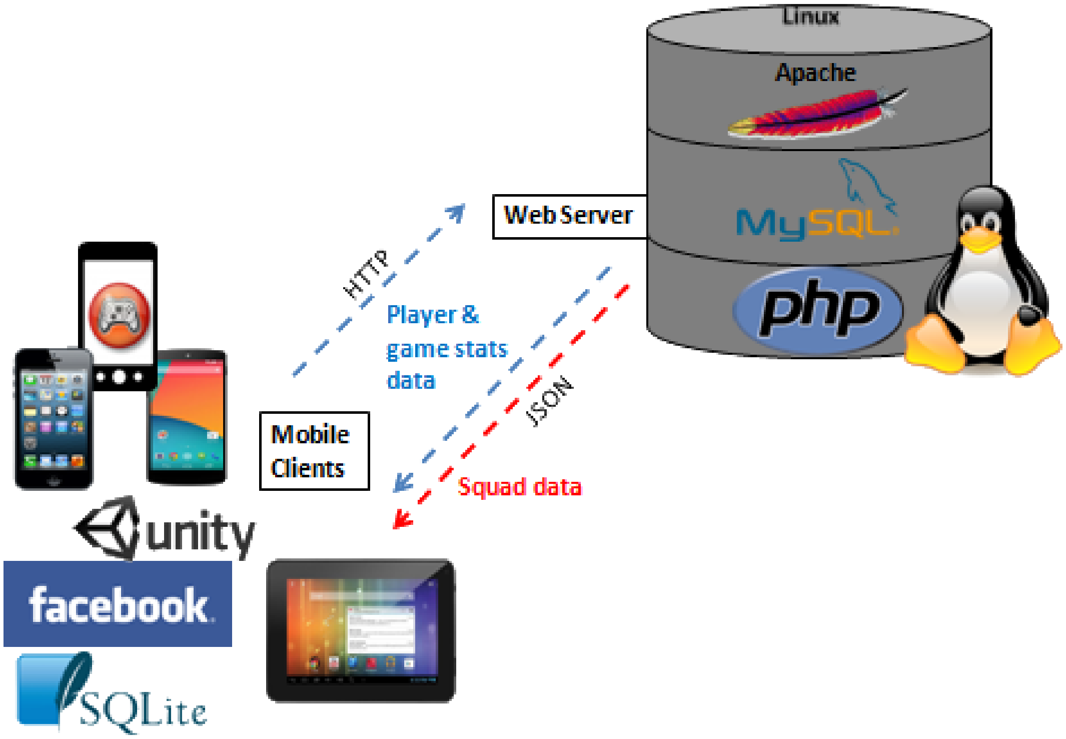


Figure 7. Architecture Diagram

Each user is the master of his/her own content via their device; a player's devices is the only one to push updates to his/her own player account, thrusters, helmets, and stats data to the server; the exception to this is any game played in which a player's character was a member of a friend's squad. In the case which a player is using multiple devices and data isn't synchronized properly, the player is given have the choice of which data to overwrite (the local copy with server data or the server data with a local copy).

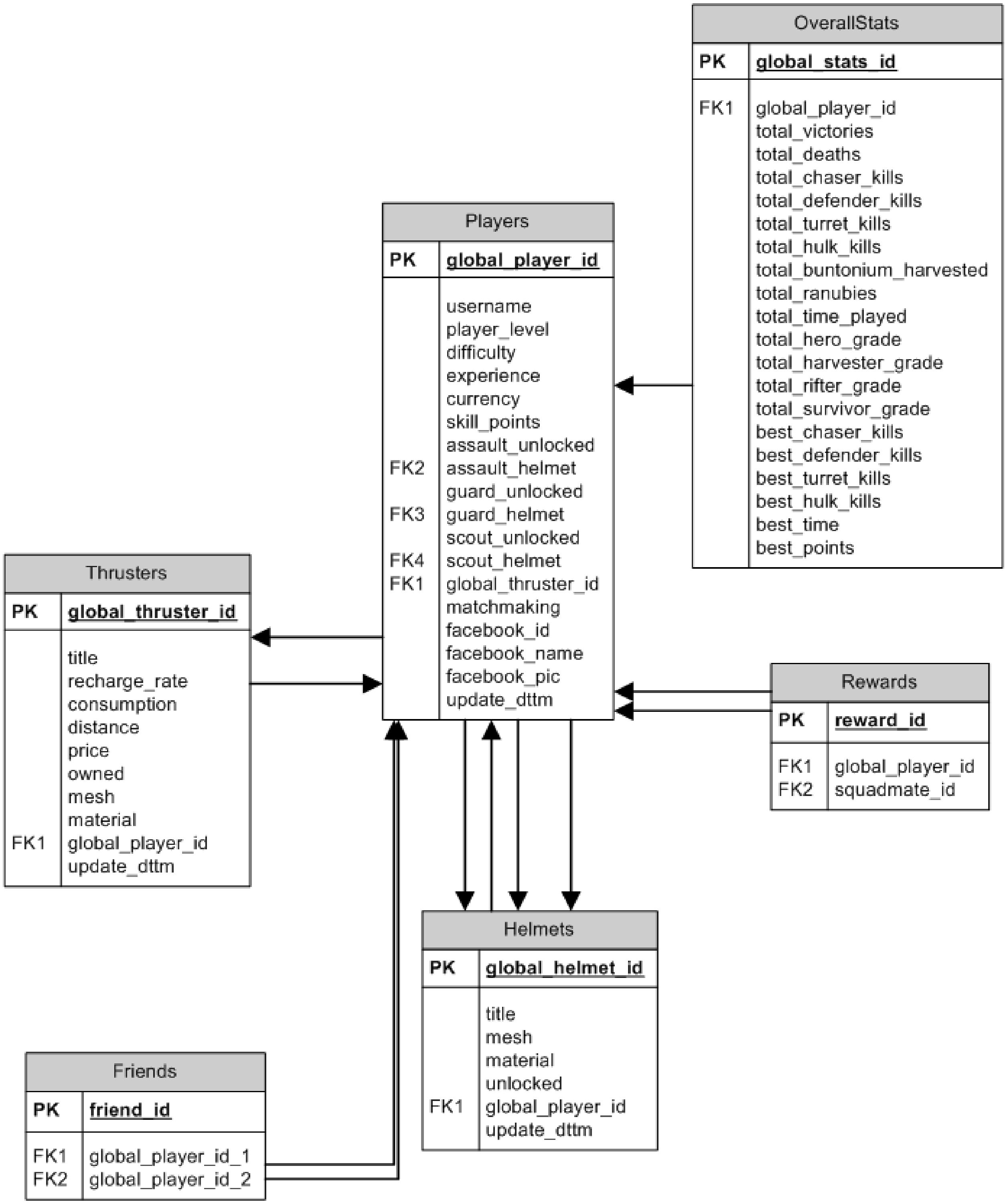
Security

To ensure data is valid and secure we validate client data on the mobile device as it is input. The data is then written into a compliant JSON string and sent to the server via HTTPS POST method to use SSL encryption.

The server validates an application key which is also sent via the HTTPS POST method before proceeding to parse the JSON string. The parsed data is escaped for XXS and SQL Injection attacks before being used in prepared statement calls to the MySQL server using a limited database user role with permissions only to SELECT, INSERT, UPDATE and DELETE tables.

MySQL Database Diagram

Creating a mobile game which shares data across multiple devices requires a centralized server to manage data. Figure 8 below is a database diagram which describes the data architecture of the MySQL database.



OverallStats	
PK	global_stats_id
FK1	global_player_id
	total_victories
	total_deaths
	total_chaser_kills
	total_defender_kills
	total_turret_kills
	total_hulk_kills
	total_buntonium_harvested
	total_ranubies
	total_time_played
	total_hero_grade
	total_harvester_grade
	total_rifter_grade
	total_survivor_grade
	best_chaser_kills
	best_defender_kills
	best_turret_kills
	best_hulk_kills
	best_time
	best_points

Players	
PK	global_player_id
	username
	player_level
	difficulty
	experience
	currency
	skill_points
	assault_unlocked
FK2	assault_helmet
	guard_unlocked
FK3	guard_helmet
	scout_unlocked
FK4	scout_helmet
FK1	global_thruster_id
	matchmaking
	facebook_id
	facebook_name
	facebook_pic
	update_dttm

Thrusters	
PK	global_thruster_id
	title
	recharge_rate
	consumption
	distance
	price
	owned
	mesh
	material
FK1	global_player_id
	update_dttm

Rewards	
PK	reward_id
FK1	global_player_id
FK2	squadmate_id

Helmets	
PK	global_helmet_id
	title
	mesh
	material
	unlocked
FK1	global_player_id
	update_dttm

Friends	
PK	friend_id
FK1	global_player_id_1
FK2	global_player_id_2

Figure 8. MySQL Database Diagram

SQLite Database Diagram

In addition to the MySQL database, *Heroes of the Rift* requires a light-weight database option for mobile devices, namely SQLite. Figure 9 below is a database diagram which explains the architecture of the SQLite database.

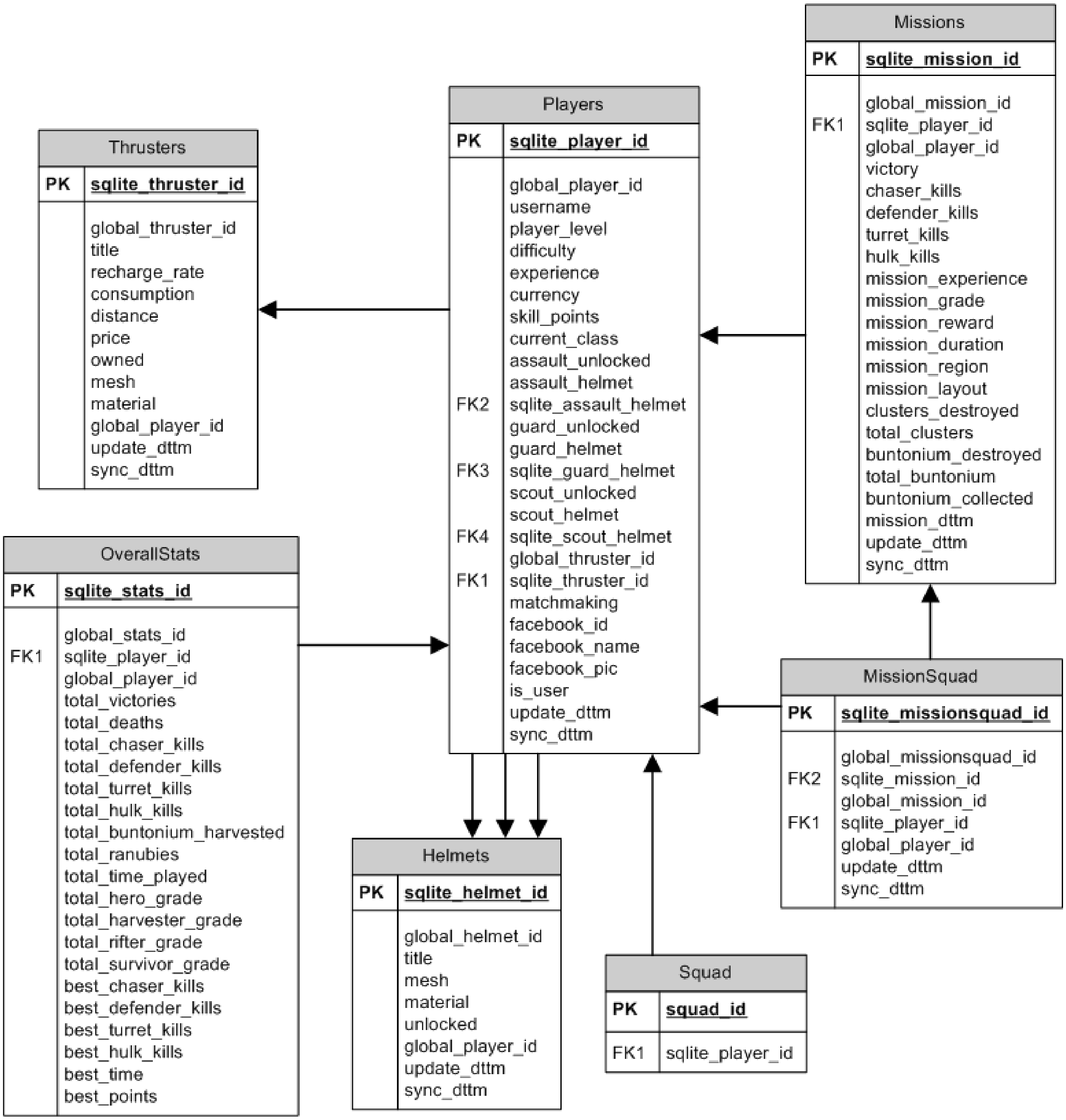


Figure 9. SQLite Database Diagram

Graphic and Audio Design

The visual style for *Heroes of the Rift* is about exaggeration and irreverence. The stylized and colorful models and environments are meant to compliment action heavy game play while maintaining a casual tone. The art design of the game complements the dynamic colors and spectacles of space. In the *Heroes of the Rift* universe, the technology appears more advanced than our own, but retain a familiar look. Thruster packs, teleportation devices and energy based weapons are commonplace. By using color highlights on player characters and bright, colorful particles, the game features an explosive, energized visual style.

Player models are stylized with exaggerated proportions. They are slightly top heavy and have expressive in-game animations to help convey information to the player. Enemy designs are primarily robotic. This allowed our team of artists to infuse each collision with particles such as sparks and explosions upon their destruction.

Heroes of the Rift takes place across several different environment locations. Each mission takes place on a “chunk” of space debris as it drifts through “The Rift”. These chunks are populated with lush jungles consisting of alien flora and fauna, ice covered planes, and magma soaked space rock. Each location features staples of our established art direction, helping to provide a layer of consistency to the game world. However, each chunk is structured to feature wildly different color pallets, making each area feel unique.

GUI Elements

Our GUI elements include heads up display images, our entire menu interface and additional 2D assets. To create these assets we leveraged Photoshop and Illustrator. The elements needed to have clean angles and spacing to allow us to convey game information to the player in an easy to consume format. In addition to the role of clearly conveying information, all of the GUI elements retain the look and feel established by our game world and its characters. Creative GUI elements that complimented our style helped create and maintain a cohesive design vision.

3D Models

To create the 3D models for *Heroes of the Rift* we used Autodesk 3DS Max 2014 and Photoshop. 3DS Max allowed us to create character, enemy and environmental models true to concept art pieces, and more importantly true to the artistic design towards which we strived. These models embrace the stylized tone of the concept art counterparts – providing colorful, exaggerated settings. Game animations are expressive and overstated to evoke excitement and generate an entertaining gaming experience. Painting and texturing the models with certain highlights and focal points helped maintain clarity on screen during the fast pace combat.

CONCLUSION

In summary, our goal is to create a fast-paced action game for the mobile platform targeting the casual gaming audience. *Heroes of the Rift* accomplishes our goal by incorporating:

- a unique “hook” in the form of our TALON combat system which simplifies controls
- social gaming through the integration of Facebook to create passive co-operative play
- minimalist RPG elements create unique game play while simplifying the complex decision-making required in traditional RPG gaming
- constant engagement of the player by our enemy spawn algorithm.

The culmination of these elements creates a simple, intuitive, and exciting experience for mobile gamers.

References

1. Summers, Nick. “Distimo: Video games were the most downloaded apps on iPad and Android in September”. *The Next Web, Inc*. November 13, 2013. *http://thenextweb.com/mobile/2013/11/01/distimo-video-games-downloaded-ipad-android-app-type-september/*