SportZ

[Project Plan]
Version 1.0

By: Lin Zixing, Fabian Wong, Chia Songcheng, Lim Sheng Zhe, Chew Poshi, Chee Zi Hoe, Hermes Lim HongJun

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	Mar 17th, 2022	Lin Zixing, Hermes Lim, Chee Zi Hoe, Chew Poshi, Ryan Chia, Sheng Zhe, Fabian Wong	First version

Table of Contents

Contents

1 Introduction

1.1 Project Overview

SportZ is a game development project that aims to encourage teenagers and young adults to exercise and play sports in Singapore. Using the endless runner genre and incorporating features of various popular sports, the game would provide an engaging and satisfying experience for the player to motivate them to play sports in real life.

1.2 Project Description and Scope

SportZ is a standalone video game software for Windows PC developed as part of the ActiveSG movement to promote healthy living as requested by Sports Singapore.

The basis of SportZ is a 2D endless runner platformer game to bring out the uniqueness of various sports and motivate players to try out various sports depicted in the game to promote healthy living. Some features of the game include:

- Variety in sports which changes up gameplay experience
- Score system to reflect player performance
- Obstacles and enemies to overcome throughout the game
- Purchasable upgrades with earnable currency to enhance gameplay experience

These features are given as an example, and more features can be found elaborated in the Systems Requirement Specification, as well as the final product.

As a standalone software, SportZ will be developed on the Unity3D engine, and no external software or interfaces are required upon the software's deployment. Development of assets such as art, animation, sound, and UI will be done on 3D applications or software. These assets will be integrated into Unity prior to actual deployment.

2 Project Organisation

2.1 Team Structure

The following is the list of executive roles, as required by CMM level 3.

- Project Manager: Lin Zixing
- Release Manager: Lin Zixing, Fabian Wong
- Software Development Group: Fabian Wong, Chew Poshi, Sheng Zhe
- UI/UX Engineer: Ryan Chia
- Software Design Group: Hermes Lim, Chee Zi Hoe
- Software Quality Assurance Engineer: Hermes Lim, Chee Zi Hoe

2.2 Roles and Responsibilities

Project Manager: Lin Zixing

- Oversees project progress
- Approves and executes project plan
- Assigns tasks and reports status of project to team members
- Manages and motivates team members
- Represents the team to the outside world

Release Manager: Lin Zixing, Fabian Wong

- Manages each release in development
- Ensure product quality for each incremental release

Tech Lead: Fabian Wong

- In charge of all technical development matters
- Coordinates code and technical components done by the development team

Developer Team: Chew Poshi, Sheng Zhe

- Implement game mechanics and various game components based on detailed design document

UI/UX Engineer: Ryan Chia

- Designs User Interface
- Ensures stability and response time of the system meet the requirements
- Creates user manual

Design Team: Lin Zixing, Hermes Lim, Chee Zi Hoe

- In charge of Project Proposal, Project Plan, System Requirements Specification and other relevant design document
- Designs game mechanics and game features

Quality Assurance Engineer: Hermes Lim, Chee Zi Hoe

- Ensures acceptable software quality
- Designs testing strategies
- Creates and manages test plan
- Verify software requirements
- Executes test procedures

2.3 Team Communication

SportZ communication channels include the following:

- Weekly meetings are held on Thursday.
- Group announcements and updates are sent through team telegram group
- Split up into development and design subgroups as necessary, to work more cooperatively on specific problems.

3 Process Definition

3.1 Lifecycle Model

SportZ intends to use the Agile Software Development model, specifically a modified version of the Scrum model.

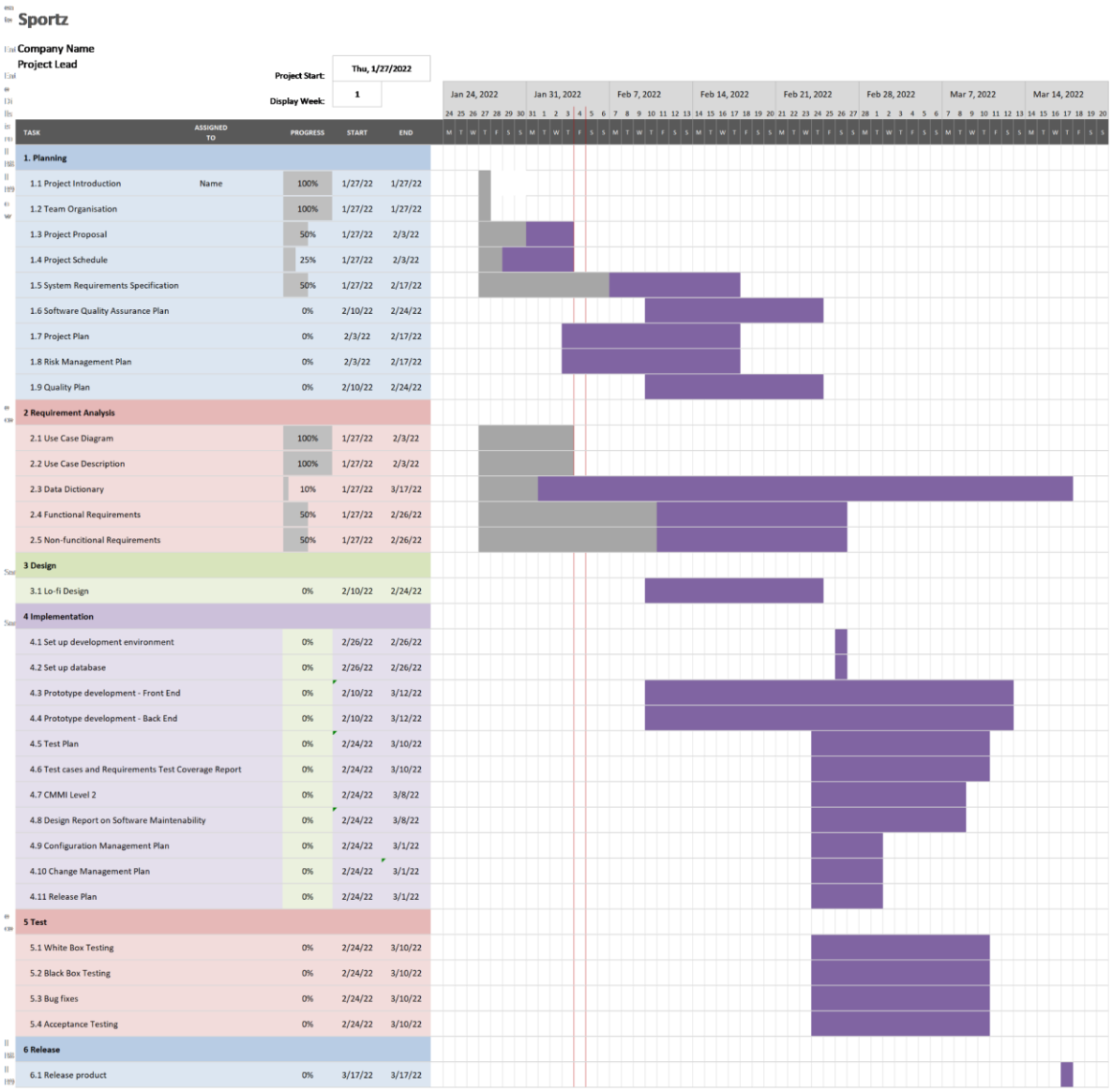
A sprint cycle of 2 weeks is utilised, with a weekly scrum meeting involving the entire group to update on status for all aspects of the project, and an internal subgroup meeting once every two days to discuss specific matters for the subgroup.

Agile is also chosen due to time constraints, which induces a need to reiterate design ideas as development progresses. This allows more flexibility over traditional SDLC models and certain ideas could be implemented on the go with minimal documentation changes. Testing will also be done regularly in various scales, either component specific testing or overall software testing.

A proper and elaborate product backlog is also designed to ensure clarity of tasks and allow for a proper measure of progress and velocity of the project. The project is expected to be delivered by the due date of 24th March with all functionalities stated in the initial design documents and necessary documentations.

4 Schedule

4.1 Activity Dependencies and Schedule



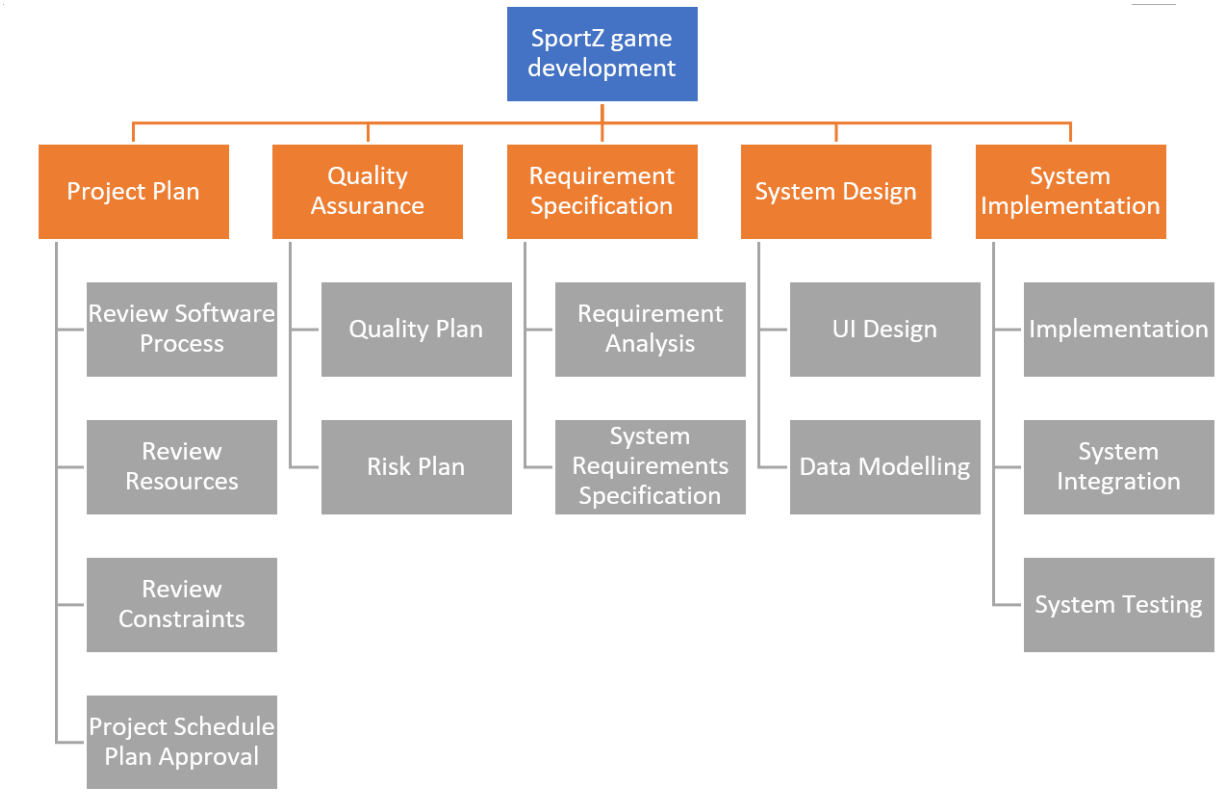
Sportz

Company Name
Project Lead

Project Start: Thu, 1/27/2022
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
1. Planning				
1.1 Project Introduction	Name	100%	1/27/22	1/27/22
1.2 Team Organisation		100%	1/27/22	1/27/22
1.3 Project Proposal		50%	1/27/22	2/3/22
1.4 Project Schedule		25%	1/27/22	2/3/22
1.5 System Requirements Specification		50%	1/27/22	2/17/22
1.6 Software Quality Assurance Plan		0%	2/10/22	2/24/22
1.7 Project Plan		0%	2/3/22	2/17/22
1.8 Risk Management Plan		0%	2/3/22	2/17/22
1.9 Quality Plan		0%	2/10/22	2/24/22
2 Requirement Analysis				
2.1 Use Case Diagram		100%	1/27/22	2/3/22
2.2 Use Case Description		100%	1/27/22	2/3/22
2.3 Data Dictionary		10%	1/27/22	3/17/22
2.4 Functional Requirements		50%	1/27/22	2/26/22
2.5 Non-funcitional Requirements		50%	1/27/22	2/26/22
3 Design				
3.1 Lo-fi Design		0%	2/10/22	2/24/22
4 Implementation				
4.1 Set up development environment		0%	2/26/22	2/26/22
4.2 Set up database		0%	2/26/22	2/26/22
4.3 Prototype development - Front End		0%	2/10/22	3/12/22
4.4 Prototype development - Back End		0%	2/10/22	3/12/22
4.5 Test Plan		0%	2/24/22	3/10/22
4.6 Test cases and Requirements Test Coverage Report		0%	2/24/22	3/10/22
4.7 CMMI Level 2		0%	2/24/22	3/8/22
4.8 Design Report on Software Maintenability		0%	2/24/22	3/8/22
4.9 Configuration Management Plan		0%	2/24/22	3/1/22
4.10 Change Management Plan		0%	2/24/22	3/1/22
4.11 Release Plan		0%	2/24/22	3/1/22
5 Test				
5.1 White Box Testing		0%	2/24/22	3/10/22
5.2 Black Box Testing		0%	2/24/22	3/10/22
5.3 Bug fixes		0%	2/24/22	3/10/22
5.4 Acceptance Testing		0%	2/24/22	3/10/22
6 Release				
6.1 Release product		0%	3/17/22	3/17/22

4.2 Work Breakdown Structure



4.3 Work Packages

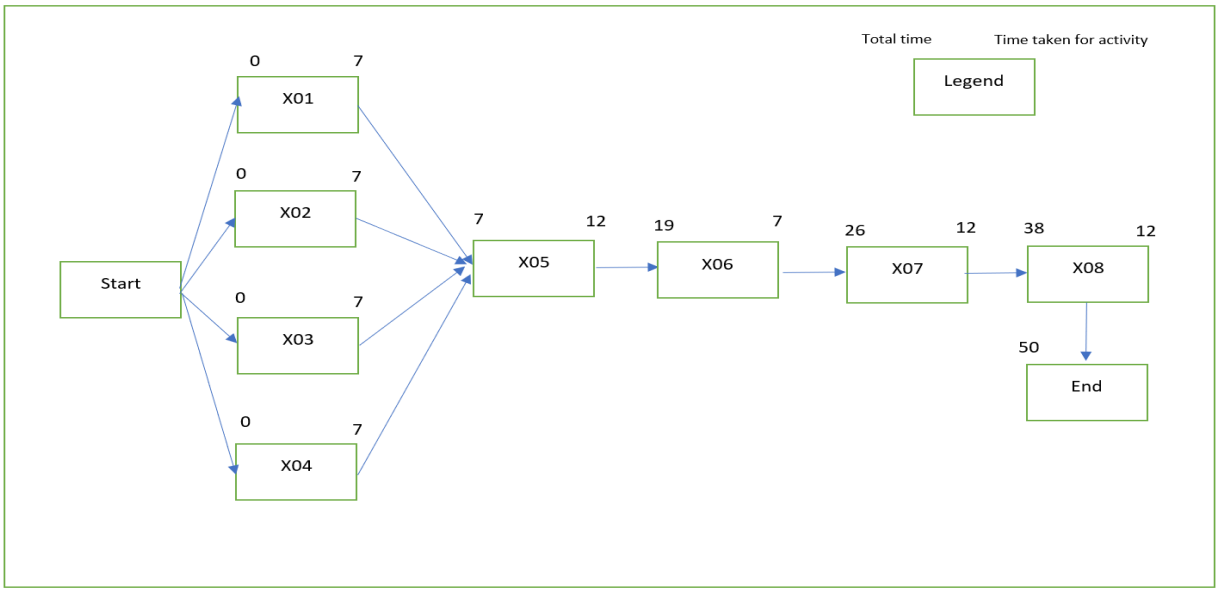
The entire project work is broken down by the important phases of the software development life cycle. They include the following:

1. Project Plan
2. Requirement Specification
3. Quality Assurance
4. User Interface
5. Technical Architecture
6. Data Modelling
7. Coding & Unit Testing
8. Integration & Quality Assurance

4.4 Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

Work Package #	Work Package Description	Duration	Dependencies
X01	Project Plan	7 days	--
X02	Requirement Specification	7 days	--
X03	Quality Assurance	7 days	--
X04	User Interface	7 days	--
X05	Technical Architecture	12.1 days	X01,X02,X03, X04
X06	Data Modelling	7 days	X05
X07	Coding & Unit Testing	16.2 days	X06
X08	Integration & System Testing	16.2 days	X07



Based on the Activity Network Diagram, the only path which is also the critical path is X01/X02/X03/X04 -> X05 -> X06 -> X07 -> X08 which takes 42 days to complete.

Below is a critical path table to visualise the slack time:

Arc (start node, next node)	Earliest Start (ES)	Earliest Finish (EF)	Latest Start (LF)	Latest Finish (LF)	Slack time
1,5	0	7	0	7	0
2,5	0	7	0	7	0
3,5	0	7	0	7	0
4,5	0	7	0	7	0
5,6	7	19	7	19	0
6,7	19	26	19	26	0
7,8	26	38	26	38	0
8,End	38	50	38	50	0

Notice that since there is only one single path we can take, that path automatically becomes the critical path. This is also the reason why there is no slack time at all since there is no additional path, we can take to reduce the time taken for the development of SportZ.

4.5 Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package.

Project	SportZ
Work Package	X01 - Project Plan (1 of 8)
Assigned To	**Chee Zi Hoe**, Hermes Lim, Lin ZiXing, Fabian Wong, Chia Songcheng, Lim Sheng Zhe, Chew Poshi
Effort	7PD
Start Date	27/01/22
Purpose	It is a series of formal documents that define the execution and control stages of a project.
Inputs	None.
Activities	The plan includes but is not limited to considerations for risk management, resource management and communications, while also addressing scope, cost, and schedule baselines. The people responsible for this work package will also be transcribing ideas brought up in the group meeting discussion into a formal report.
Outputs	A written document of the Project Plan Introduction.

Project	SportZ
Work Package	X02 - Requirement Specification (2 of 8)
Assigned To	**Hermes Lim**, Chee Zi Hoe, Lin ZiXing, Fabian Wong, Chia Songcheng, Lim Sheng Zhe, Chew Poshi
Effort	7PD
Start Date	27/01/22
Purpose	To establish a common understanding between the customer and the software project team so that customers' requirements can be addressed better by the project team.
Inputs	Customers' requirements.
Activities	Identify what is essential and to be included in SportZ. Plan the deliverables according to the user's needs.
Outputs	A written document of the requirement specification

Project	SportZ
Work Package	X03 - Quality Assurance (3 of 8)
Assigned To	**Chee Zi Hoe**, Hermes Lim, Lin ZiXing, Fabian Wong, Chia Songcheng, Lim Sheng Zhe, Chew Poshi
Effort	7PD
Start Date	27/01/22
Purpose	This establishes and maintains set requirements for developing our product. This is also meant to increase customer's confidence and the project team's credibility, while improving work processes and efficiency.
Inputs	Project Plan work package 01.
Activities	SportZ will adopt an Agile methodology, where each step in the work process is approached as a sprint. This makes our game development highly adaptive.
Outputs	A written document of both a quality assurance plan and a risk management plan.

Project	SportZ
Work Package	X04 - User Interface (4 of 8)
Assigned To	**Lim Sheng Zhe,** Hermes Lim, Chee Zi Hoe, Lin ZiXing, Fabian Wong, Chia Songcheng, Chew Poshi
Effort	7PD
Start Date	27/01/22
Purpose	To build the user interface between the system and the customer, make it easy to use, and friendly to the customer.
Inputs	User information
Activities	To get the user information, user request, display the dialog between system and user, display the result of request.
Outputs	Prototype of User Interface

Project	SportZ
Work Package	X05 - Technical Architect (5 of 8)
Assigned To	**Chew Poshi,** Hermes Lim, Chee Zi Hoe, Lin ZiXing, Fabian Wong, Chia Songcheng, Lim Sheng Zhe
Effort	12PD
Start Date	03/02/22
Purpose	To design high-level architecture design.
Inputs	Project Plan Work Packages (01 to 04 inclusive).
Activities	High-level design entails defining the software system's architecture and identifying the various components and how they are inter-related to and interactive with each other. Designers also need to decide on the software and hardware infrastructures, such as what operating system on which the software is built, the language used to implement the software, etc. Design topics including maintainability, portability, and reusability will be addressed here as well.
Outputs	High-Level Design and Architectural Specification.

Project	SportZ
Work Package	X06 - Data Modelling (6 of 8)
Assigned To	**Chia Songcheng**, Hermes Lim, Chee Zi Hoe, Lin ZiXing, Fabian Wong, Lim Sheng Zhe, Chew Poshi
Effort	7PD
Start Date	15/02/22
Purpose	To build the project's database.
Inputs	Project Plan Work Packages (01 to 06 inclusive).
Activities	Analyse the data flow relationships, entity relationships.
Outputs	A written document of the data modelling.

Project	SportZ
Work Package	X07 - Coding & Testing Unit (7 of 8)
Assigned To	**Fabian Wong**, Hermes Lim, Chee Zi Hoe, Lin ZiXing, Chia Songcheng, Lim Sheng Zhe, Chew Poshi
Effort	12PD
Start Date	22/02/22
Purpose	To implement the system as per the requirements specification and other associated documents. This work package includes such additional activities as preliminary unit testing.
Inputs	Project Plan Work Packages 07.
Activities	Programmers will implement the modules according to the design specifications noted in the Specification document.
Outputs	Source code and header files.

Project	SportZ
Work Package	X08 - Integration and System Testing (8 of 8)
Assigned To	**Lin ZiXing,** Hermes Lim, Chee Zi Hoe, Fabian Wong, Chia Songcheng, Lim Sheng Zhe, Chew Poshi
Effort	12PD
Start Date	6/03/22
Purpose	To identify and fix logical and syntactical errors produced during the implementation of the System and setting up drivers and stubs to see how the module responds to various inputs. Black box testing and white box testing might be conducted to check for logical errors. All the testing procedures will be documented in the Test Plan report. If problems are found, they will be noted and fixed at the earliest possible time.
Inputs	Project Plan Work Packages 08.
Activities	The Integration testing team may try to simulate how a user might interact with the system. Similar to Unit Testing, Integration Testing may require the development of stubs and drivers as well, but this is more geared towards the higher (overall system) level. Testers may also examine issues such as system performance and integrity. Heuristics assessment plays an essential role in this work package, as intelligence components will define eventual system success.
Outputs	A test reports.

5 Project Estimates

5.1 Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

5.1.1 Unadjusted Function Points

SportZ supports the following proposed functions:

User:

- Gameplay of SportZ
- Computation of high score
- Interaction with shop elements
- Accumulation of rewards (coins)

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation of the complexity is as follows:

Rating Inputs:

- Game Controls: Jump Button (Space bar, left click of mouse)
- User's interaction with game interface

Rating Outputs:

- Collision detection and response
- Game object animation
- General game state

Rating Inquiries:

- View shop inquiry
- View high score inquiry

Rating Logical Files:

- User settings preference
- High score files
- Upgrade files
- Coins files

Rating Interfaces:

- Not applicable to our software as there are no interfaces that will direct data to external systems and applications. Our game runs fully on unity and no external dependencies on external libraries, servers are used.

Summary of above analysis:

Element	**Complexity**	**Detail**
Inputs	High	Game Controls
	High	User Interaction with game interface
Outputs	High	Collision detection and response
	High	Game Objects
	High	General game state
Inquiries	Medium	View shop inquiry
	Low	View high score inquiry
Logical Files	Medium	User settings preference
	Low	High score files
	Low	Upgrade File
	Low	Coin Files

Calculation of Unadjusted Function Points:

Characteristic	**Low**		**Medium**		**High**	
Inputs	0	× 3	0	× 4	2	× 6
Outputs	0	× 4	0	× 5	3	× 7
Inquiries	1	× 3	1	× 4	0	× 6
Logical Files	3	× 7	1	× 10	0	× 15
Interfaces	0	× 5	0	× 7	0	× 10
Unadjusted FP	24		14		33	
Total=L+M+H	71					

5.1.2 Adjusted Function Points

Influence Factors	**Score**	**Detail**
Data Communications	4	Application is more than a front-end, and supports more than one type of teleprocessing communications protocol.
Distributed Functions	2	Distributed processing and data transfer are online and in both directions.
Performance	5	Response time or throughput is critical during all business hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
Heavily used	2	Some security or timing considerations are included.
Transaction rate	0	Daily peak transaction period is anticipated.
On-line data entry	3	More than 30% of transactions are interactive data entry
End-user efficiency	2	Four to five of the efficiency designs are included
On-line data update	2	Online update of major internal logical files is included.
Complex processing	1	Any one of the complex components
Reusability	4	The application was specifically packaged and/or documented to ease re-use, and the application is customized by the user at source code level.
Installation Ease	0	No special considerations were stated by the user *but* special setup is required for installation.
Operational Ease	5	Effective start-up, back-up, and recovery processes were provided, but no operator intervention is required (count as two items).
Multiple sites	0	User requirements do not require considering the needs of more than one user/installation site.
Facilitate change	3	Flexible query and report facility is provided that can handle complex requests, for example, *and/or* logic combinations on one or more internal logical files (count as three items).
Total score	33	
Influence Multiplier = Total score × 0.01 + 0.65 = 33 × 0.01 + 0.65 = 0.98		
Adjusted FP = Unadjusted FP × Influence Multiplier = 71× 0.98 = 69.58		

Scoring (0 – 5)
0 = No influence
1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

5.1.3 Lines of Code

According to Capers Jones[1] statistics, each Function Point requires 53 lines of code if the application is implemented using Java. Our project is mainly developed using unity codes in C++, however the code bears more resemblance to Java. Therefore, Java's LOC/FP is used instead of C++.
Therefore, we have: **Lines of Code** = 63.58 FP × 53 LOC/FP = **3369.74 LOC**

5.2 Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- Working days include 5 days in a week.
- Effort = Size / Production Rate = (3369.74 LOC) / (39 LOC/PD)[2] = 86.4036 PD = 86PD
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (86.4036) $^{1/3}$ = 13.2627 Days
- Initial schedule = 13.2627 Days / 5 days a week = 2.6525 Weeks
- Team size = 86.4036 PD / 13.2627 D = 6.5148 P = 7 Persons
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 86.4036 PD × 8 hours = 691.2287 PH = 692 PH

[1] http://namcook.com/how-srm-works/Case%20Example-6.pdf

[2] Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

5.2.1 Distribution of Effort

1990's Industry Data	Work Package	Distribution	Estimates
Preliminary Design 18 %	Project Plan	9%	62.28
	Requirement Specification	9%	62.28
Detailed Design 25 %	User Interface	7%	48.44
	Technical Architecture	11%	76.12
	Data Modelling	7%	48.44
Code & Unit Testing 26 %	Code & Unit testing	21%	145.32
	Online Documentation	5%	34.6
Integration & Test 31 %	Integration & Quality Assurance	31%	214.52
	Extrapolated total effort		692
	2% for project management		13.84
	3% for contingency		20.76
	Total effort		726.6

These duration estimates assume that each team member works an equal amount on any given work package.

5.3 Cost Estimates

Hardware:

Programmers' workstations:

4 - Aftershock Workstation (The Focus)	Total $8000
AMD Ryzen 5 5600X	
8GB KLEVV Performance	
Gigabyte X570 Aorus Elite	
Zotac GTX 1650 OC	

Designers' workstations:

3 - Aftershock Workstation (The Focus)	Total $4500
AMD Ryzen 5 3600	
8GB KLEVV Performance	
Gigabyte X570 Aorus Elite	
Zotac GTX 1650 OC	

Common use:

1 - Hp Printer	Total $3000

Software:

GNU, Apache, or Other Free Licence-based Software:

Unity3D	$0.00
Piskelapp	$0.00
Royalty Free Music	$0.00

Software Licence Provided by Third Party:

Microsoft Office 2000	$0.00
Microsoft Project 2000	$0.00
Products ESTIMATE Professional	$0.00

Other Resources:

Staff:

7 Employees with 726.6 working hours with $18.00/hour	$13078.80

Stationary:

Paper, photocopying and other miscellaneous cost	$50.00

Total: $28628.80

The above-mentioned hardware, software and other resources pertains to the various equipment that will be utilised by the development team of SportZ. SportZ's hardware and software responsibilities relate only to our own development needs to accomplish the project we have been asked to complete, and which has been described in the introduction section of this document. SportZ will also demonstrate the completed product.

6 Product Checklist

The plan is that the items listed below will be delivered on the stated deadlines.

Project Deliverable	**Deadline**
Project Proposal	8th Feb 2022
Use Case Model	8th Feb 2022
System Requirements Specification	22nd Feb 2022
Quality Management Plan	22nd Feb 2022
Project Plan	1st March 2022
Risk Management Plan	1st March 2022
Demo: Prototype for simple demo	9th March 2022
Configuration Management Plan	21st March 2022
Change Management Plan	21st March 2022
Release Plan	21st March 2022
CMMI Level 2 Definition	21st March 2022
Test Plan	21st March 2022
Test Cases and Requirements Test Coverage Report	21st March 2022
Final Product	21st March 2022

7 Best Practice Checklist

Practice	✓
Ensure that documentation is done and is in a standardised format.	✓
Check for ambiguity, completeness, accuracy, and consistency. The requirements need to be met and the requirement documentation must contain a complete functional specification.	✓
Complexity management is one of the major challenges. ● Minimise interfaces between modules, procedures and data. ● Minimise interfaces between people, exponential communication cost. ● Avoid fancy product functions and designs. Functionalities need to meet ● the customers' requirements.	✓
The manager must have good communication with his or her employees; require developers to make code available for review; review design for appropriateness	✓
Planning for continuous change: ● All manual designs, test, source code should have revision numbers and dates, revision history comments, change marks to indicate the changes. ● New revisions should be approved before being made and checked for quality and compliance after being made. ● Use a configuration management system and make processes. ● Required maintenance.	✓
Avoid underestimating by obtaining accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance.	✓
Code reviews is an efficient method to find software defects. Plan and manage code reviews between team members.	✓
Software testing will use both black box and white box testing . It will involve unit, functional, integrating and acceptance testing.	✓
Set smaller milestones to enable better monitoring, better control and better risk management, and in general mitigate incidents in a more controlled way. Team members should meet to set these mini-milestones and align them with the big milestones to meet deadlines and reduce the delays that may arise due to interdependencies of assigned tasks.	✓
Ensure that team members are aware of one another's responsibilities to avoid double work.	✓

8 Risk Management

Besides the general risk management, the following risks have been identified for our project:

Underestimation of Project Size

- Impact Severity: High
- Probability: 25%
- Impacts: Depending on the stage at which the changes occur, could range from needing to update the requirements documentation to needing to do a complete redesign.
- Risk Reduction: Be rigorous in eliciting requirements. Make customers aware of potential repercussions of requirement changes.

Changing specifications

- Impact Severity: High
- Probability: 15%
- Impacts: Delay in finalising the specification will push the schedule for all following stages of the project.
- Risk Reduction: Monitor progress of specification carefully.

Retention of employees

- Impact Severity: Extreme
- Probability: 5%
- Impacts: Happens when staff leave halfway before the completion of the project. There would be more work for remaining employees, and any specialised skills or knowledge would be lost.
- Risk Reduction: Offer benefits and incentives to staff.

Cancellation of project

- Impact Severity: Extreme
- Probability: 1%
- Impacts: All work and resources will have been wasted if customer cancels the project.
- Risk Reduction: Keep in close contact with customers. Ensure that they have some market research indicating a demand for this product.

Negligent in monitoring of project status

- Impact Severity: Moderate
- Probability: 5%
- Impacts: Poor quality of monitoring can lead to the true project status largely unknown amongst the development team.
- Risk Reduction: Regular updates should be documented and made known during meetings for any further followup actions that are needed.

Unrealistic Schedules

- Impact Severity: Moderate
- Probability: 5%
- Impacts: Project milestones cannot be met, resulting in the delay for project launch date
- Risk Reduction: Proper project planning and management is required to ensure that the development team are able to meet the deadlines realistically.

9 Quality Assurance

The project will achieve quality assurance by following the standard set by the company. The specific procedures and details shall be provided in the Quality Plan. Specific test procedures and details shall be provided in the Module/System Test Plan. In addition, testing of SportZ shall make use of two testing methodologies:

- **Unit Testing** involves testing system components individually.
- **In-Place Testing** involves testing of the whole system as a unit.

Furthermore, these methodologies will be used to test two important aspects of the game environment:

- **System Function** will be tested to ensure that software flaws are eliminated, and
- **Algorithmic Function** will be tested to ensure that the logic for gaming controls and avatar, and game play are precise and error-free.

SportZ's methodology makes broad use of realistic test cases. Detailed test data is an important part of the final project delivery. SportZ will validate code and gameplay of our game using realistic scenarios. Our development team will test the project to ensure users will have a smooth gaming experience. Gaming environments such as game user interface, player controls and interactions with game objects will also be tested to ensure that the features in the game are operating as intended.

10 Monitoring & Control

Many procedures are required to be able to successfully monitor the progress of a software project. Some of the most important are:

Quantitative measurement of resource consumption: Estimates of the development resource requirements, primarily in terms of human resources, can provide a quantitative measurement of project progress through the measurement of product backlog completion. Some of the Quantitative Analysis that can be adopted in SportZ include:

- Sensitivity Analysis
- Simulation

Identification of major project risks: Early identification of major risks to the project allows for placement of preventative measures before problems can develop. Major risks have been identified in the Risk Management section of this document, along with the measures being taken to avoid them. Some of the identification tools SportZ can adopt include:

- Brainstorming
    - Our team will attempt to generate ideas or find a solution for a specific problem that was brought up.
- Interviewing
    - A fact-finding technique our team can use where we will interview our faculty professors that have similar project experience to identify potential risks in our project.
- Risk Register
    - A document the team will use to contain the results of various risk management processes.
    - Helps document potential risk events and related information

Qualitative Risk Analysis: This helps our team to monitor SportZ by assessing the likelihood and impact of identified risks to determine their magnitude and priority. This is critical as risk factors change from time to time. We are able to use to following tools and techniques:

- The Top Ten Risk Item Tracking
    - Helps the team identify risks as well as maintain an awareness of risks throughout the life of SportZ
    - It lists the current ranking, previous ranking, number of times the risk appears on the list over a period of time, and a summary of progress made in resolving the risk item.

Regular reviews of project progress: Throughout the duration of the game development project, the team shall meet weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing.

Assigned members in SportZ will establish a periodic review of the top ten project risk items

Timeline Planning and task decomposition: This document outlines an estimated timeline for the project. A reasonably accurate timeline can be assembled by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition can assist in task assignment and balancing. Throughout the implementation phase, these subcomponents can allow for fine-grained measurement of progress.

Our team has come up with a Gantt chart for this purpose as seen in section 4 of the project plan. Estimates are also aided alongside by the work breakdown structure in this document.