Project Plan

Group 9
James Theron - 1901870
Alon Mizrahi - 1405583
Ashish Jugpall - 2273877
Tenisha Moodley - 2105982

1. Game Pitch

Imagine playing a game similar to billiards but on a mini golf course. This is the core mechanic of this game concept. This game takes the player through puzzle-packed courses which take inspiration from a mini golf game, using a similar level of agency that one would have in a game of billiards. The game fancies the use of these game elements in order to make silly yet strategically designed game courses, encouraging players to test and explore the numerous interesting ways in which they can sink the objective ball into the pocket at the end point of each stage.

2. Product Introduction/Overview

Project Outcomes and Requirements
This project requires a single-player, physics based, digital game concept that utilises the Iterative design process in order to design the game according to the constant changes the game implements throughout development. The outcome of such a process should result in a complex-dependent game that focuses it's depth through Data design, Feedback design, Level design and Feedback loops rather than through numerous complicated game elements.

Project Management Methodology
Project plans are used to help developers communicate with all of the members involved with the project and further help developers perceive project risks, budgets and dependencies in order to reduce risks, manage the work, and guide themselves from the beginning to the end of the project. The project management will therefore follow a value system concerning game elements and tasks, in order to prevent discrepancies within the design of the project.

Start and End Dates
24th May - 18th June which allocates less than 4 weeks to complete the game and the reflection of the iterative game process.

3. Process

The development process of the game will start by developing a working prototype which allows the player to operate the basic hitting mechanics which the game intends to be built upon. This should allow for basic playtesting of the primary game mechanic. Once the player is able to move and apply force to the game pieces in a way that can allow the game to function in its most basic form, work on the ability mechanics can begin. The development of said ability mechanics should be accompanied by the implementation of the environmental game objects as they are intended to function in a symbiotic manner with one another. The obstacles which the ability mechanics are used by the player to overcome, should each be tested individually. For instance, the placing of walls within any of the levels should be tested in how it provides the player the opportunity to use the given ability mechanics like the ability to break walls or move through them. This is intended to forge meaningful application of the game's secondary mechanics. Once all the planned mechanics have been implemented in a working prototype which allows for full playtesting of the level designs, external playtesting may begin to a further extent in order to collect data regarding the game in its initial prototype phase. This data will be taken into account when finalizing the tuning of mechanics as well as improving any issues which come to attention from the playtesting process. Playtesting will also help gauge the level of communication brought across by the game and should help improve anything that requires more clarity.

Once the gameplay is at a satisfactory stage, polishing of the final product can begin. This will include the implementation of user interface elements for menu navigation as well as general touching up of the visuals.

Documentation regarding how the process will be carried out on a week to week basis can be seen on the scheduling found on the appendix.

4. Milestones

Working Initial Prototype

1. Program the ball controller to allow the player to apply a force to hit the balls with the desired amount of power and direction. A line renderer should denote both power and direction.
2. Create prefabs for the various game objects that will act as obstacles and implement them in a basic level format to be playtested.
3. Implement initial ideas for the basis of each level design individually.
4. Program ability ball mechanics to behave as intended.
5. Program the success state to occur when the player reaches the goal region with the objective ball.
6. Program a fail state to occur when the player does not complete a stage within the given limitations.
7. Implement basic forms of communication for implemented mechanics.

Iterating on Initial Prototype

8. Finalize the level designs.
9. Finalize the tuning of the variable constants which influence the behavoural physics of both the ball hitting mechanics as well as the environmental aspects.
10. Finalize ability mechanics to work smoothly within their respective level environment.
11. Use playtesting data to finalize any issues found with initial prototype mechanics and level designs and improve communication for anything that is found to be unclear.

Polishing the Final Product

12. Implement further user interface elements which communicate the limitations of the given level. This will also include UI which allow the player to operate mechanics within the game like switching between the ability mechanics.

13. Implement the menu user interface elements which allow the player to quit the game, enter a specific level, or potentially view a tutorial screen (tutorial may be included within the initial levels).
14. Apply finishing details including custom fonts, materials, textures.

5. Task Breakdown

The task breakdown table below describes the logical order of milestone completion and the set of tasks needed to achieve each milestone. The order of tasks is done in the order of dependencies. The tasks further down the timeline are dependent on the previous tasks. Each task is also given a category of either ‘need’, ‘want’, or ‘dream’. These categories denote the importance of each task completion to achieve each milestone. The first set of milestones are aimed at achieving a practical, working prototype, that will allow for effective early on playtesting.

Table showing the task breakdown of each milestone and each tasks implementation necessity status

Milestone	**Need/Want/ Dream**	**Task**
Program the ball controller to allow the player to apply a force to hit the balls with the desired amount of power and direction. A line renderer should denote both power and direction.	Need	Create ball objects and add physical properties.
	Need	Create a controller that allows interaction with the balls.
	Need	Add a visual trajectory line of balls.
Create prefabs for the various game objects that will act as obstacles and implement them in a basic level format to be playtested.	Need	Create each game object. This includes the level floor, obstacles(scenes) and other interactables.
	Want	Create extra level objects.
Implement initial ideas for basic level designs.	Need	Use level prefabs to create a basic level for testing.
	Want	Create more complex levels.
Program ability ball mechanics to behave as intended	Need	Create the ability ball and add it’s physics controller.

	Need	Create various ball abilities.
	Need	Implement a UI which allows the player to switch between the current ability assigned to the ability ball.
	Want	Create extra ball abilities.
	Want	Add detailed visual communication to Game objects
	Dream	Add animations and sounds to Game objects
Program the success state to occur when the player reaches the goal region with the objective ball	Need	Create a visual UI element that shows the player has failed.
	Need	Allow the player to move to the next level.
	Want	Allow the player to repeat the level.
Program a fail state to occur when the player does not complete a stage within the given limitations.	Need	Create a visual UI element that shows the player has failed.
	Need	Add reset/repeat UI level button.
	Dream	Add animation to the UI element.
Implement basic forms of communication for implemented mechanics.	Need	Add visual communication and feedback to objective ball and ability ball
	Need	Add visual distinction between each level object
	Want	Add sound to game objects.
	Dream	Add animations to gameobjects.
Finalize the level designs	Need	Create a set of levels that are diverse using a multitude of game objects
	Need	Create tutorial, or beginning levels that introduce the player and guide them through the beginning of play.
	Dream	Have a minimum of 10 levels

Finalize the tuning of the variable constants which influence the behavoural physics of both the ball hitting mechanics as well as the environmental aspects.	Need	Use data design integrated into game objects to change their behavior according to playtesting feedback.
	Need	Use playtesting data to update communication design on level elements.
Finalize ability mechanics to work smoothly within their respective level environment	Need	Use playtesting data to update ability mechanics.
Use playtesting data to finalize any issues found with initial prototype mechanics and level designs and improve communication for anything that is found to be unclear.	Need	Make changes to communication design, level design, game objects and mechanics as needed.
Implement further user interface elements which communicate the limitations of the given level. This will also include UI which allows the player to operate mechanics within the game like switching between the ability mechanics.	Need	Such as if ball ability has been used, then that colour ball should no longer be available to the player in that level.
Implement the menu user interface elements which allow the player to quit the game, enter a specific level, or potentially view a tutorial screen (tutorial may be included within the initial levels).	Need	Create an in-game menu screen.
	Want	Create a start menu.
Apply finishing details including custom fonts, materials, textures.	Need	Use play testing data to make any changes to game objects, level design, communication design, and mechanics
	Want	Add animations to UI and other game objects

6. Feature List

The game should aim to have about the same features in each iteration developed, with only minor additions and/or changes with good reason. Using all the game mechanics, level and data designs, each iteration of the game should possess the following features:

- A Top-down physics-based puzzle game developed for the PC format.
- The game will include a minimum of 3 levels.
- A main menu, pause menu and quit application function however a tutorial page may be optional.
- Either mouse and keyboard support or gamepad controller support.
- The player can hit the ability ball and use it's mechanics to traverse the course.
- An aim indicator allowing players to see the direction of where they are about to hit the ball.
- A minimum of 2 ability ball mechanics which the player may choose to use.
- Ball physics that allow the ball to act as the developer would want.
- A set of environmental game objects including things like walls, switches, force fields, death zones and moving surfaces.
- A backend system which tracks the player's efficiency in the completion of levels.

7. Scheduling

The schedule[1] will follow a weekly cycle that is geared towards milestone achievement, testing, and iteration. Figure 1 below describes a generic week of the schedule. Each week has a 'design', 'implementation', 'review', and 'testing' phase. The schedule is designed to encourage iteration and re-design, based on information gained from play testing.

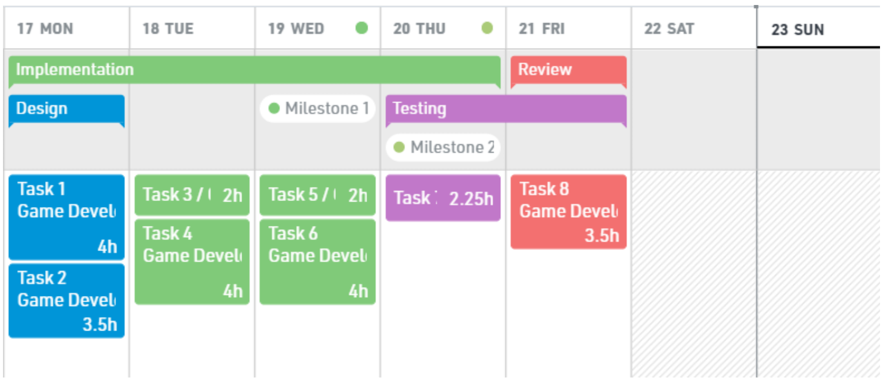


Figure 1: A generic weeks' structure with example phases, milestones and tasks

A testing period will be established after each milestone has been implemented, this testing period will be used to encourage iteration and redesign, allowing the designer to 'follow the fun'. There is no activity scheduled for Saturday and Sunday, this time will be reserved for overcoming any delay due to any uncontrollable variables, these are detailed in section '8. Risks'.

The full schedule can be seen in the appendix, subsection A.

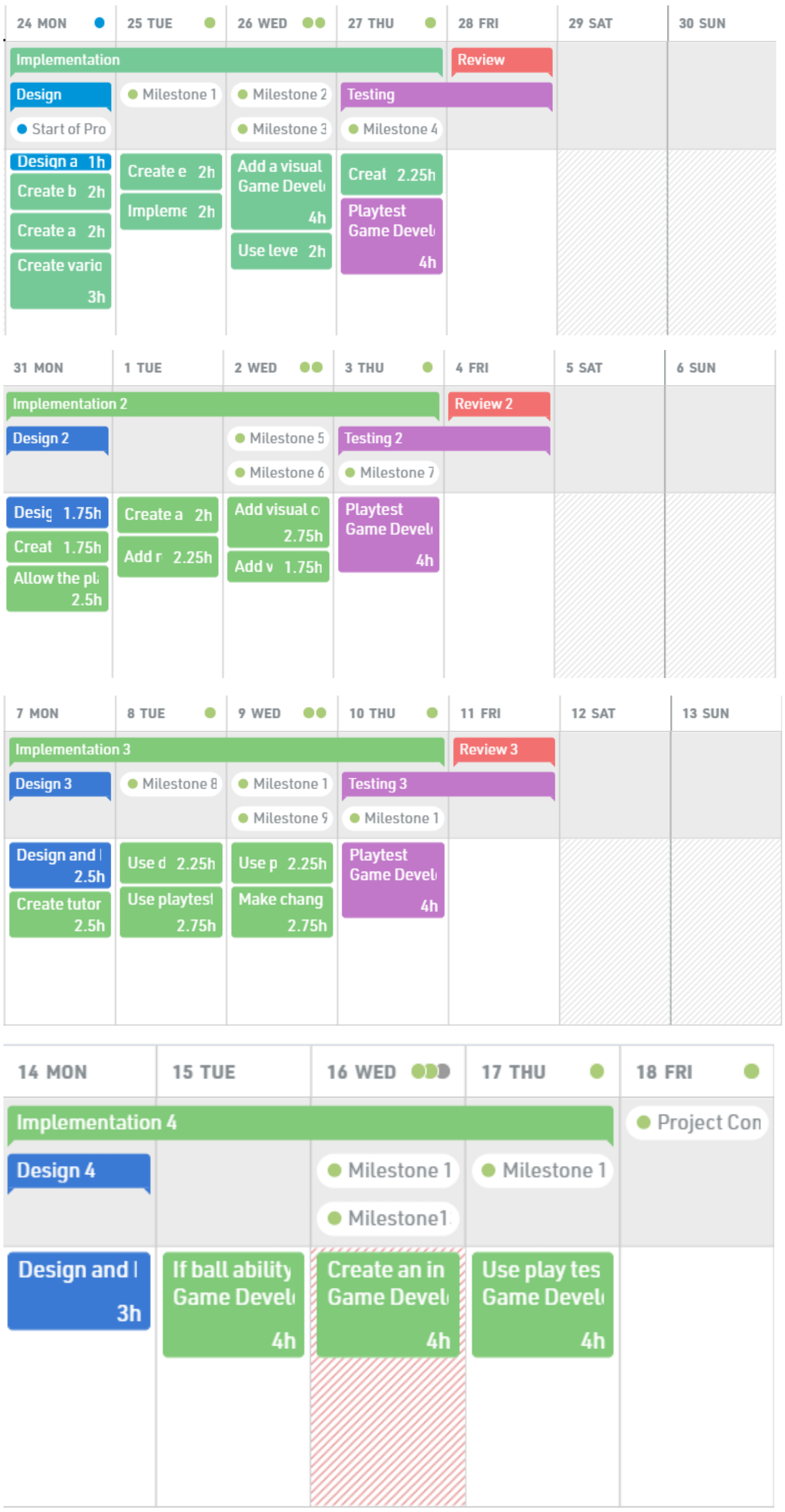
8. Risks

A Table showing the Risks that could take place during the development of this project.

Risks	**Primary Level**	**Secondary Level**	**Tertiary Level**	**Why**
Load Shedding	**X**			Primary level risk because load shedding is a high possibility in South Africa
Illness		**X**		The risk of illness could have various effects on productivity depending on the severity of the illness. It could potentially be serious because of COVID-19 or other serious illnesses, however it could not be serious and therefore not that big of a risk
Internet Failure (Unrelated to electricity failure)			**X**	Tertiary level risk because it is unlikely to happen, but still has a possibility with electricity failures affecting the working ability of the internet

9. Appendix

A. Full schedule for the duration of the project



24 MON ●	25 TUE ●	26 WED ●●	27 THU ●	28 FRI	29 SAT	30 SUN
Implementation				Review		
Design	● Milestone 1	● Milestone 2	Testing			
● Start of Pro		● Milestone 3	● Milestone 4			
Design a 1h	Create e 2h	Add a visual Game Develo 4h	Creat 2.25h			
Create b 2h	Impleme 2h	Use leve 2h	Playtest Game Devel 4h			
Create a 2h						
Create vario 3h						

31 MON	1 TUE	2 WED ●●	3 THU ●	4 FRI	5 SAT	6 SUN
Implementation 2				Review 2		
Design 2		● Milestone 5	Testing 2			
		● Milestone 6	● Milestone 7			
Desig 1.75h	Create a 2h	Add visual c 2.75h	Playtest Game Devel 4h			
Creat 1.75h	Add r 2.25h	Add v 1.75h				
Allow the pl 2.5h						

7 MON	8 TUE ●	9 WED ●●	10 THU ●	11 FRI	12 SAT	13 SUN
Implementation 3				Review 3		
Design 3	● Milestone 8	● Milestone 1	Testing 3			
		● Milestone 9	● Milestone 1			
Design and 2.5h	Use d 2.25h	Use p 2.25h	Playtest Game Devel 4h			
Create tutor 2.5h	Use playtest 2.75h	Make chang 2.75h				

14 MON	15 TUE	16 WED ●●●	17 THU ●	18 FRI ●
Implementation 4				● Project Con
Design 4		● Milestone 1	● Milestone 1	
		● Milestone1		
Design and 3h	If ball ability Game Devel 4h	Create an in Game Devel 4h	Use play tes Game Devel 4h	

10. References

[1] Float.com. (2021). *Float - Resource Planning & Management Software*. [online] Available at: https://www.float.com/ [Accessed 23 May 2021].