

DETAILED GAME SPECIFICATION: PAUL'S ASTRO GOLF ADVENTURE [PAGA]

COURSE: COMPUTING MACHINERY II [COMP 2659], WINTER 2022

INSTRUCTOR: PAUL POSPISIL

AUTHORS: JOE SPURWAY

SOREN BRADLEY

LAST MODIFIED: TUESDAY, JANUARY 18, 2022

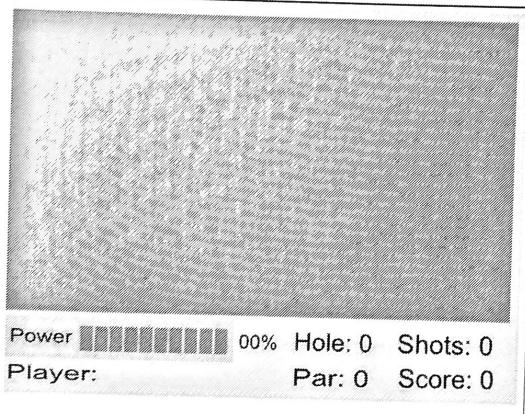
The player play one hole at a time.

no need to state this. - OBVIOUS

I. General Game Overview

This document outlines the functional components of the 2D 9-hole mini-golf game [Mini Golf] from a top-down perspective. Mini Golf provides the user with the ability to control the angle of a putting ball (char 'A' or 'D') and power of their shot (char 'W' or 'S') and take their shot by pressing the enter key (char 'IFT'). To signify the angle's position changing relative to the ball, a directional line will be used that implies the direction the ball will travel. Following this, we will use a graphical power bar to visualize the output power of the shot. Obstacles are placed on the map in the form of surface terrain changes and portals/redirects. Surface terrain changes, changing the speed of the ball lowering the power of the moving ball. Portals teleport the ball to another area on the map. The ball leaves the portal with the same power and angle that it entered the portal with. The perspective of the game takes a bird's eye view of course, where the user can see the whole map and all its features to accurately calculate their next shot. The goal of the game is to get the user's ball into the putting-hole at a defined position on the map in the least amount of turns possible.

Commented [JS1]: [High-level game concept summary paragraph goes here.]



II. 1-Player: Game Play Details

START SCREEN

Commented [JS2]: [First sample screenshot goes here. Remember to properly cite all images obtained from other sources.]



The game will open with a menu screen displaying the mini-golf game logo followed by entry buttons for the two game modes – 1-player and 2-player and an exit button. There will be accompanying music to accompany a dynamic star like background that moves behind the menu scene. From this point the user will be sent to a level selection menu – or jump straight in to continue procedurally through a 9-hole course.

GAME START STATE

A 1-player game session begins with the first hole- opening to two major screen sections, the map box area, and the stat box area. As seen below the stat box will display the current hole (map number 1), the par for the current hole, and the number of shorts taken so far (0) along with the power and angle of the shot both set to the default value of (0).

GAME RULES

Players must complete each hole in number order without skipping any hole. Players use the options provided (angle/power) to define a hit of the golf ball from the tee area and straight into the hole. The objective is to make it into each hole in as few moves as possible. Every stroke taken counts as a point, and each level as a defined par for an optimal completion time. The player with the fewest strokes at the end of the round wins.

END SCREEN

Ending the game via completing all available holes takes you to an end screen with high scores and an option to return to the menu, quit or play again.

cute!

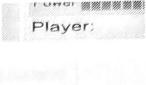
I suggest starting
with 1 level

angles - starting angle
- changes by how
much for each
key press
- Add change direction
how?

power - same as above
obstacle details

OBJECTS

Object or Object Type Name	Properties	Behaviors	Graphical Image [when available]
Screen	Map Box Stat Box Dimensions: 640x400 px	Arranges objects to form UI. <i>display</i>	
Stat Box	Position integers - x coordinate: 0 px - y coordinate: 118 px Power Bar Box Total Score Box Map Number Box Par Box Shot Box Player Box Size: 118 x 400 px	Contains stat objects <i>display</i>	
Power Bar Box	Position Integers - x coordinate: 0 px - y coordinate: 118 px Power Box Power text - Size: 64 x 75 px Power percentage - Size: 64 x 25 px Type: string Size: 64x200	Contains power box Updates Power percentage Displays power text and power percentage <i>Power 0% 10 levels</i> <i>3 levels</i> <i>the screen?</i>	
Power Box	Position Integers - x coordinate: 64 px - y coordinate: 118 px Power segment	Updates to current power setting Displays current power setting	

	Size 64 x 100 px		
Player Box	Position Integers - x coordinate: 0 px - y coordinate: 54 px Type: Integer Max Players: 2 Size 44 x 100 px	Displays Player count	
Total Score Box	Position Integers - x coordinate: - y coordinate: Integer Max Score integer 450 (9 holes max 50 shots each) Size: 44 x 100 px	Displays score Updates score	
Hole/Map Number Box	Position Integers - x coordinate: - y coordinate: Type: Integer Max Map Integer: 9 Size 64 x 100 pixels	Displays current hole Updates to new hole	
Par Box	Position Integers - x coordinate: - y coordinate: Type: Integer Max Par Integer: 7 Size: 44 x 100 pixels	Displays par of hole Updates par	
Shot Box	Position Integers - x coordinate: - y coordinate: Type: Integer Max score: 50 Size 64 x 100 pixels	Displays shots taken Updates shots taken	
Map Box	Position Integers - x coordinate: 0 px - y coordinate: 640 px Start Position - x coordinate: - y coordinate: Size 512 x 380 pixels	Displays current map Displays ball position and ball angle indicator Displays terrain Definition Box's in places	

	Array of Terrain Definition Box Ball Hole Array of portals Array of Walls Ground	Displays Hole Displays Ground Displays Walls Displays portals	really just boundaries
Ball	<p><i>* Size</i></p> <p><i>* Moving</i></p> <ul style="list-style-type: none"> - Position Integers <ul style="list-style-type: none"> - x coordinate: dynamic - y coordinate: dynamic - Top Point Position = $x + 7, y + 8$ - Bottom point Position = $x + 7, y - 15$ - Left point Position = $x + 0, y - 7$ - Right point Position = $x + 15, y - 7$ - Center point Position = $x + 7, y + 7$ - Integer angle (in steps of 15 degrees) [I.E., int 1 = 15° & int 24 = 360°] <ul style="list-style-type: none"> - Max angle value: 24 - Min angle value: 0 - Integer power unit (in steps of 10) <ul style="list-style-type: none"> - Max power value: 100 - Current Velocity (pixel person second): Integer <ul style="list-style-type: none"> - Max value 10 - Min value <u>-10</u> ? - Ball Directional Pointer - Size: 16 x 16 pixels 	Displays ball at current x/y position Velocity is set based on a power and angle function Can collide with walls, terrain, and Hole Creates the velocity (power per pixel) via the power input value.	move.
Ball Directional Pointer	<ul style="list-style-type: none"> - Position Integers <ul style="list-style-type: none"> - x coordinate: center x - y coordinate: center y - Ball angle Integer <ul style="list-style-type: none"> - Max angle value: 24 - Min angle value: 0 - Size: 16 X 128 pixels 	Displays the direction that the ball will go based on the angle set by user.	
Ground Definition Box	<ul style="list-style-type: none"> - Position Integers <ul style="list-style-type: none"> - x coordinate: - y coordinate: - Integer Friction Coefficient <ul style="list-style-type: none"> - Standard Value: 2 	Is the foundation of the playable area that the ball moves on. Friction Coefficient is subtracted from	

going to need to

8-way motion
0 - 1.0 ?
want to avoid real values & trig functions
~~WAY TOO SHOW~~

?

6

	Size: 496 x 364 pixels	the balls currently velocity every 10 pixels	
Terrain Definition Box	Position Integers <ul style="list-style-type: none"> - x coordinate: dynamic - y coordinate: dynamic Integer Friction Coefficient <ul style="list-style-type: none"> - Standard Value: 4 Size: 128 x 128 pixels	When ball enters Terrain box the balls velocity will be effected by the friction coefficient Friction Coefficient is subtracted from the balls currently velocity every 8 pixels	
Portal	Position Integers <ul style="list-style-type: none"> - x coordinate: dynamic - y coordinate: dynamic Position of connected portal <ul style="list-style-type: none"> - x coordinate: dynamic - y coordinate: dynamic Size: 32 x 32 pixels	When the ball crosses the portal boundry the balls x/y coordinates will be replaced with the partner portals as their velocity and angle remain unaffected.	
Small Wall Box	Position Integers <ul style="list-style-type: none"> - x coordinate: dynamic - y coordinate: dynamic Size Horizontal: 16 x 64 Pixels Size Vertical: 64 x 16 Pixels	Acts as a boundary and obstacle for the ball on the map.	
Medium Wall Box	Position Integers <ul style="list-style-type: none"> - x coordinate: dynamic - y coordinate: dynamic Size Horizontal: 16 x 128 Pixels Size Vertical: 128 x 16 Pixels	Acts as a boundary and obstacle for the ball on the map.	
Large Wall Box	Position Integers <ul style="list-style-type: none"> - x coordinate: - y coordinate: Size Horizontal: 16 x 380 pixels Size Vertical: 380 x 16 pixels	Acts as a boundary and obstacle for the ball on the map.	
Hole	Position Integers <ul style="list-style-type: none"> - x coordinate: - y coordinate: - Center Hit Boolean Size: 32 x 32 pixels	Game continues until Hit Boolean is True.	

how are these
different?
if not different
what is the point?

ball - X collision

wall
hole
portal

CONDITION-BASED (CASCADED) EVENTS

Event Name	Triggering Input Event	Description
Get ball in hole	Ball box x/y position equal or within bounds of hole hitbox.	Map score is added to the total score. Hole session is ends and the user moves to the next map or finishes.
User Takes Shot	'LF' [enter] will initiate the shot sequence with parameters defined by sleeted angle and power.	Ball travels at a velocity defined by the powers inputted power unit * 10 pixels to get power per pixel metric.
User changes power	'w' will increase the power setting 's' will decrease the power setting	Power bar will change to reduce or increase the shown bar segment via a viable.
User changes Ball angle	'a' rotates the direction anti-clockwise 'd' rotates the direction clockwise	The Ball direction pointer moves to the angel specified by the user defined ball angle.

what is this?

HYPOTHETICAL GAMING SESSION

Upon opening the game, the user is greeted with an opening menu screen with an accompanying musical jingle. This splash screen presents the user with options of a one or two player mode along with an exit button. After the user selects the game mode required the music continues to a hole selection menu. The first hole will be presented and the screen will be updated with the first mini gold map loaded.

The user can manipulate the mechanics of the game through the 'w', 'a', 's', 'd' and 'LF' characters on the keyboard. By altering the angle (char 'a' or 'd') and power (char 'w' or 's') then taking shot (char 'LF' or enter) the user picks their route through the map to achieve optimal completion by comparing shots against the current par for the hole. As the ball moves around the course the interactions it makes with its surroundings will comprise the challenge of the game. The users can interact with features to aid or hinder them like terrain changes that will slow the ball and magical teleporting portals that will move the ball into favorable map positions.

Upon the user's completion of a hole the next map will be loaded in for the user. Upon completion of all 9 holes, the user will either be sent back to the home splash screen or within the two-player mode, a score comparison will be presented.

III. 2-PLAYER: GAMEPLAY DETAILS

Objectives and Rules Modifications

The two-player version of mini golf resembles similar mechanics to that of a single player game with the addition of a second ball on the course. The screen will be identical for both players with the addition of both users' stat box data being visible to both users, (or if possible, independent to each user) and will limit the user to input only when it is their turn. The hitbox of each ball will now also need to account for interaction between ball and ball, time permitting this would stop or react to this event.

Modifications

needs beefed out with course layout objects.

layout - tee?
- hole?
- obstacles?
- too vague
actual game play

Rules

The player who gains the least points across the total available holes wins.

The player who gains the most points across the total available holes loses.

Gameplay

Assuming the same, shared interface we will display both players' stat box on the display.

Assuming different screen outputs we will display each player specific stat box to the relevant screens.

Both players' golf balls will be shown on either screen.

Ball to ball interactions will need to be handled appropriately.

IV. SOUND EFFECTS

Sound Effect Name	Event Triggering Playback	Description [Linked]
Song (opening splash screen)	User opens the program	The Police's 'Walking on the Moon'
Song (gameplay)	User starts the game	Peter Schilling's 'Major Tom'
User shot	User shoots the ball	Splonk Async
Enter Hole	Ball enters hole	Yasssss Sync
Enter portal	Ball crosses portal	Pooooof

Background Music

Walking on the Moon sets the scene for this space fairing mini-golf game with its funky tones and chill galactic themes, any keen Astro-golfer would love to dance to. Following this, the song Major Tom shoots us into a fun action-packed multi-level mini-golf game.

V. ADDITIONAL FEATURES (TIME PERMITTING)

Inclination and Declination

For each terrain box that makes up a play space we hope to create a velocity modifier to create the effect of increase or decrease in the pitch of the surface the ball travels across. From this we hope to visually represent these different modifiers via shading and lines to signify their representative class - shallow incline, regular incline, or a steep incline.

Level selection menu to access each of the 9-holes independently.

For each level we hope to create an accessible menu icon on a splash screen to join a new game in any hole independently without playing through all 9.

Jumping

For both the balls maneuverability, and opportunity for added in map features - the implementation of a 'jump' feature allowing the ball to move into a raised state to jump over things like walls, gaps, and environmental dangers would be beneficial.