

Mobile Interaction: Cargo Balance

Background:

Cargo Balance is a unique skill-based game that incorporates real-time weather data into gameplay when the game starts. The game itself is aimed at a younger demographic, having a simple game loop of tutorial, select ocean, select difficulty, and aim for the highest score you can. The real-time data provides varied experiences on a selection of oceans where each can feel different depending on the time. The main goal is to last as long as possible within the degree of error lines to prevent your ship's cargo from falling. Once it does, the game over screen will prompt users to view the scoreboard to compare and gauge for later runs.

The app works well on many mobile devices, the simple tilting control and visual feedback being the basis of gameplay helps to improve the overall usability and experience for players.

Potential design opportunities include having improved visual feedback, along with various icons and sounds to help users prepare for a large wave which could cause cargo to fall if not accounted for. Furthermore, technical opportunities could include utilising larger wave data sets or additional API calls to dynamically update the waves as appose to one call at the start. This method could provide a greater opportunity to stand out in the market as similar applications do not include live features actively affecting gameplay.

After some research, I was unable to find many similar applications, however two stood out as having a similar basis with waves and cargo in one and tilting gameplay in another. These include Ship Balance Puzzle, and Balance Maze: Tilt Game, although both games have a similar basis, they do not prompt competitiveness or survival in the same way. This leaves a potential gap for Cargo Balance to stand out as it incorporates different features used in multiple games to provide an engaging tilting survival experience rather than a puzzle.

Description and Justification:

Activities and Structure

Cargo balance uses basic activities, intents and functions, with the gameplay operating on a simple loop. The first page (Figure 1) provides a starting point with button links to a tutorial, starting and a scoreboard. On start the gameActivity file hosts the gameView, where pausing, settings and game-over are handled. The scores are managed by scoreboardActivity, which displays the last score and the top three scores based on each difficulty. This is to avoid having the score loading whilst the user is unable to see it on screen. The app fits well into the wider android ecosystem as standard activities and system services are used, along with API, sensors, vibration and audio being common features on many devices. All sensors and audio are stopped when the app is running in the background.

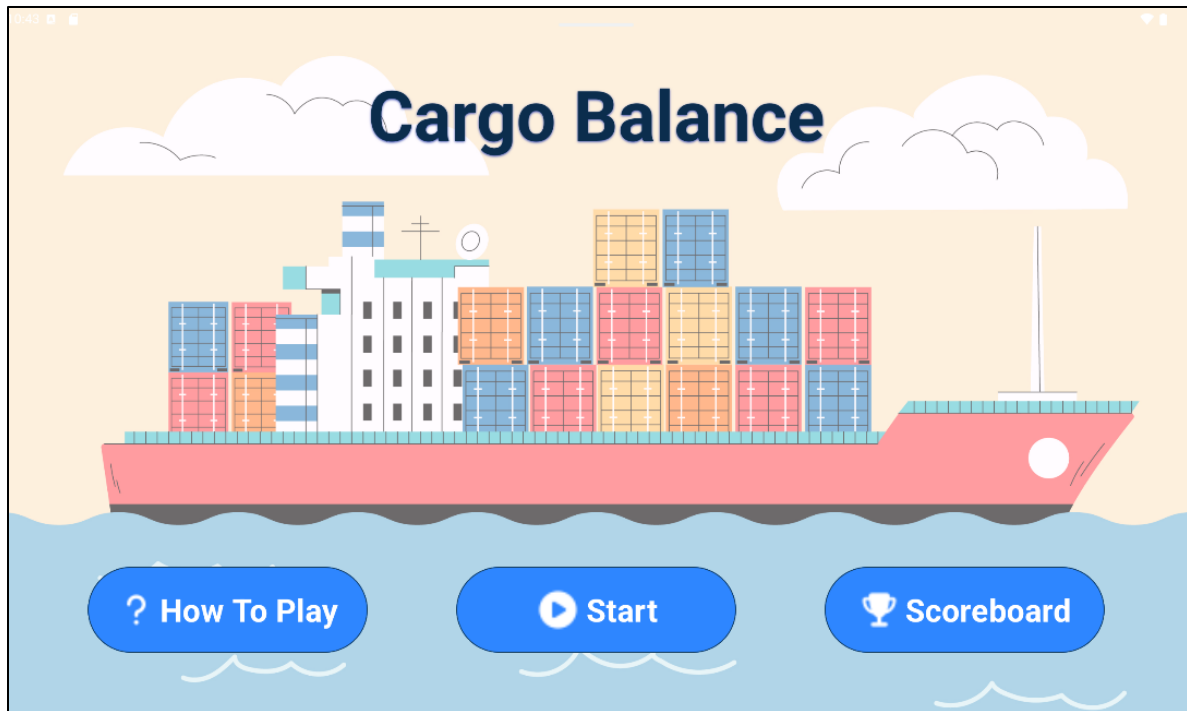


Figure 1 – Main-Menu

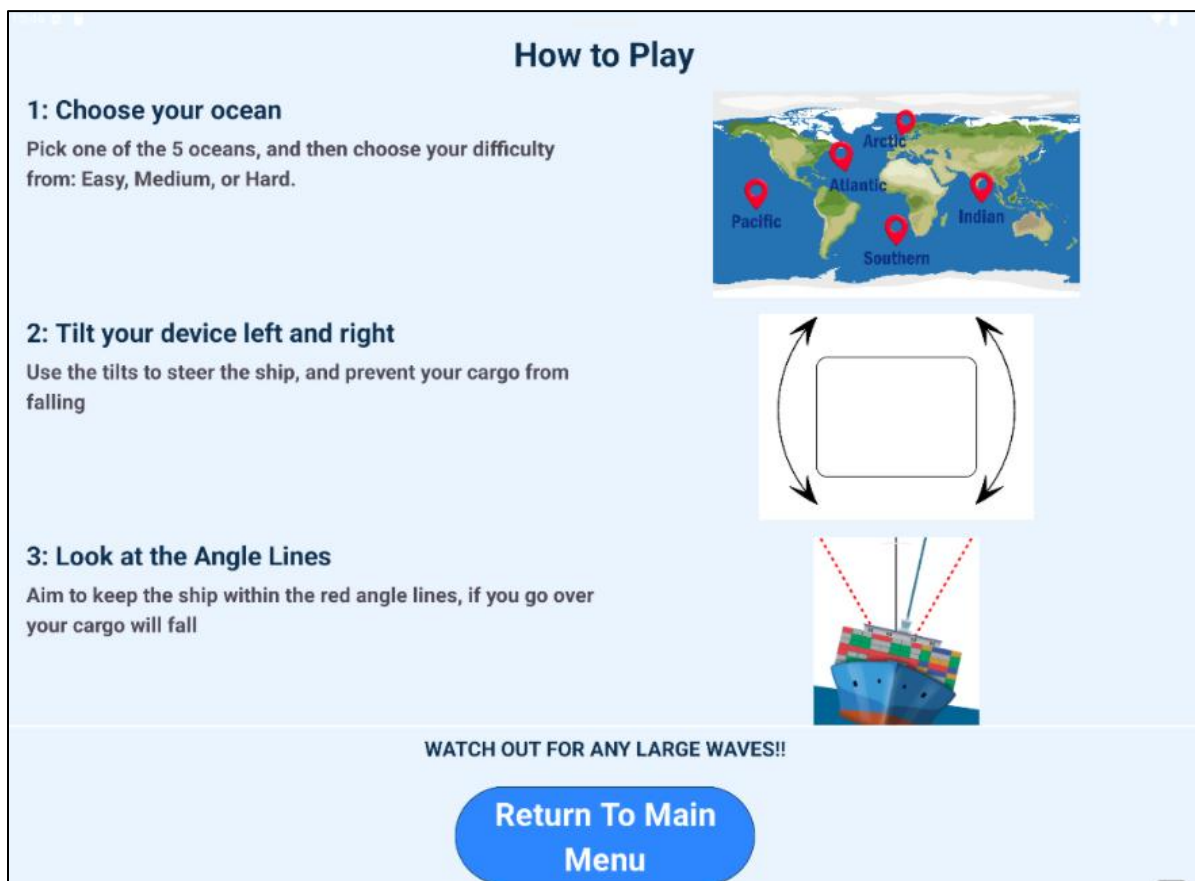


Figure 2 - Tutorial

Gameplay Loop (SurfaceView)

Gameplay is drawn using a surfaceView, which enables the use of Canvas to have constant drawing, making the game look smoother. The main game is handled on a secondary thread which provides updates dynamically and draws each frame. For the drawing, each loop finds the time since the last frame, gets updates on waves and ship movement, locks the canvas with SurfaceHolder, draws, then posts the finished frame to the surfaceView display. To prevent a previous bug of jumping movement and device lag, the time between frames is limited, in addition to the thread sleeping briefly to average out around 60 fps.

Wave System and API

The waves in Cargo Balance use a sine wave function added to a normal drawn line to create a constant rocking motion on the screen. With variables like wave height and period, the degree of tilt in the waves are controlled as well as how often the motion repeats, imitating the waves of an ocean. The values for height and period are constrained based on tests to where the ocean values can be so extreme that the wave height exceeds the screen height. A starting countdown along with an intro wave function are used to improve useability so players aren't thrown in to an unbeatable challenge. The intro wave is set on every loop where the waves gradually increase to the normal wave height of the ocean over a set time. As hinted in the how to play page (Figure 2), there are chances of large waves, these occur every three wave cycles, which varies based on period, also having a 90% increased height than a normal one. To prioritise competitiveness, large waves are given little indication and are aimed to be challenging. A small one-shot vibration is played at the start, along with a slight water colour change. The wave heights and periods are retrieved once the user presses the ocean they would like to play using the Open-Meteo Weather API, with hard coded height and period values if the API fails.

Gravity Sensor and Ship Smoothing/Weighting

The tilting controls in Cargo Balance utilise the android device built-in gravity sensor, which allows the user to tilt their device to steer the ship towards the centre lines. The sensor takes initial readings of the users tilt as the basis for where the ship tilts from, however this approach can cause inaccuracy if the game is started whilst the device is held at an awkward angle when starting the game. When the initial tilt is established, the gravity values are converted into a tilt amount which is then scaled to the target steering angle of where the ship is intended to be. A previous bug in development was how the gravity controls was specific for landscape, although this was fixed as directionality works with both landscape and reverse landscape without affecting tilting in any way. A small dead zone was added to make the tilting feel more valuable, as small movements started to jitter the ship in an unnatural way. To avoid the ship snapping to the target angle, the amount of steering that can be done is constrained to avoid fast movements causing jumps to the tilt angle. Lastly, to smooth movement, the shipWeight variable along with angle and tilt speed cause the ship to feel less responsive.

Ship Drawing + Guidance lines

Initially I used a rectangle to mimic the ship and how it would move, however once this worked and I swapped to using the ship bitmap, it would not. To get around this issue, I use the bitmap scaled onto a rectangle which also provides consistency across multiple devices. The ship is then moved using Canvas rotation, so the ship angle matches the calculated angle from the wave and steering logic. With the main aim being to prevent cargo from falling, angle lines have

been added to help improve the usability of the game. Two outer dashed lines show the degree of error (Figure 3), where they change based on the easy, medium and hard difficulty levels. The angle lines are drawn from the centre of the ship, so they line up correctly regardless of screen size, this also clearly identifies the angle limits for the game.

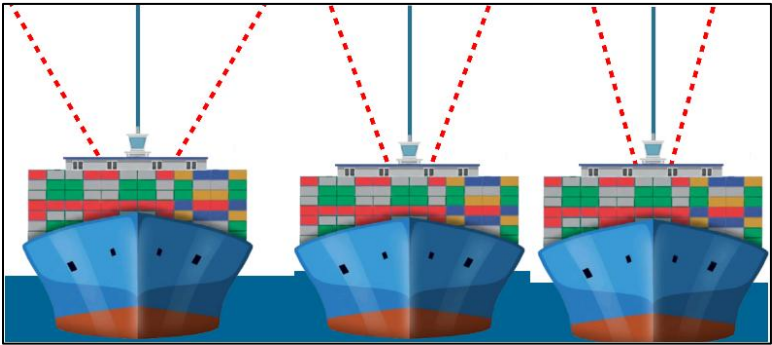


Figure 3 - Angle Lines

Settings, Permissions and Ethics

Cargo Balance makes use of SharedPreferences data file storage to save values that stay saved when app is closed and reopened. The main premise of this is storing the top three scores in each difficulty to be then displayed on scoreboard (Figure 4). However, using the same system, user settings for audio, muting vibration and angle line colours, can all be saved in the same way as users would expect to keep their settings in and out of the app. The permissions the app requests are internet and vibrate, having internet is useful as it allows for live data but not entirely required for gameplay as fallback values are in place if the API fails. The vibrate is required to play the small one-shot vibration whilst a large wave starts. Security and ethical concerns are manageable as the app does not collect any personal data and has no login or location tracking. The app is reliable and can run well without an internet connection as medium difficulty values have been put in place. Additionally, all sensors and audio are stopped when the app is minimised reducing battery drain.

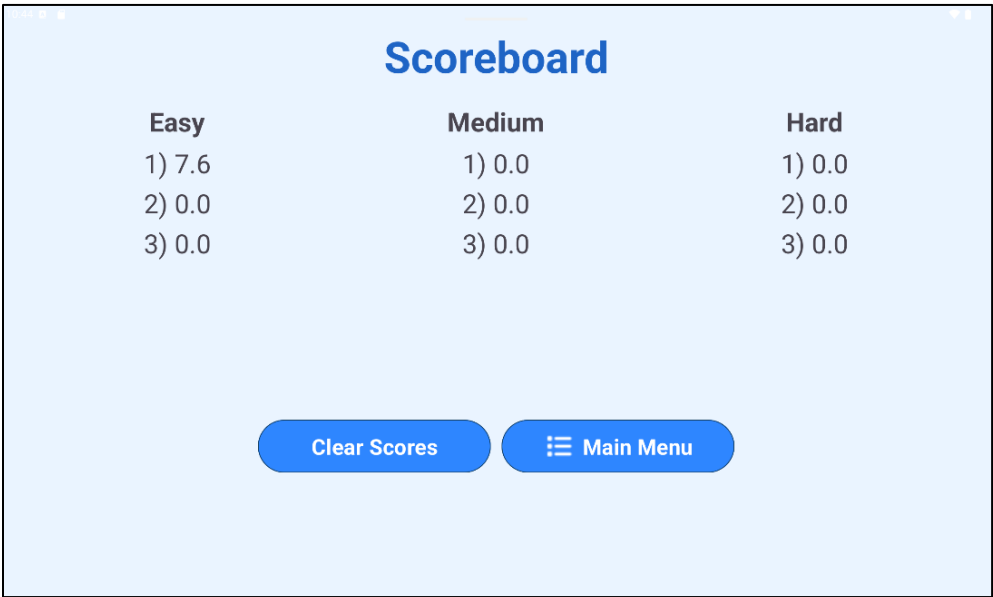


Figure 4 - Scoreboard

Critical Reflection:

The aims of the project were simple, create a skill-based game with ties to oceans and ships. In short, Cargo Balance fits these aims by linking real world data to a tilting control system and a repeatable gameplay loop. The main gameplay is easy to understand and the whole app provides information for new players, and challenging/scoreboard aspects for existing players. The API integration works well; the wave height and period values affect gameplay and enables players have different experiences depending on the ocean they select.

The gameplay works as initially thought out, with ship movement feeling intuitive and natural, with guidance lines to make it useable for players. The wave system creates a challenge for players with changes depending on the ocean.

Some features of the app were not implemented; this was mainly due to technical limitations but also time constraints. The main premise was to have a second mode where the movement of a car swaying or speeding up could affect gameplay in some way, however this exceeded my technical capabilities in the time I had left.

If looking back at the app, I would focus on fixing bugs, such as the device tilt issues, and audio playing in pause menu occasionally. Additionally, I would focus on implementing new features to improve the user experience, having indicators for large waves, achievements, options to change the ship, and weather icons next to each ocean showing difficulty based on height and period data. I would next aim to improve the waves by adding more API calls from different locations within the ocean boundary to have waves dynamically update throughout gameplay.

Word Count: 1641

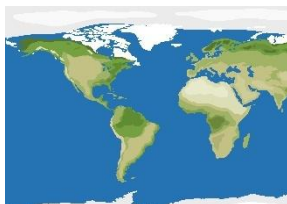
Reference List:

Parsley Games (2024) Ship Balance Puzzle, APKPure. Available: <https://apkpure.com/ship-balance-puzzle-and-arcade/com.shipbalance.puzzle/amp> Accessed: 13/01/26

shuumatsu.programmer (2026) Balance Maze: Tilt Game, Google Play. Available: https://play.google.com/store/apps/details?id=work.androidkaihatsudomain.k014_balance_game Accessed: 14/01/26

Open-Meteo (2026) Weather API, Open-Meteo. Available: <https://open-meteo.com> Accessed: 13/01/26

Description of Asset	Source	Location on App	License/Permission
Font Awesome Icons: ?       	https://fontawesome.com/icons/question?f=classic&s=solid https://fontawesome.com/icons/circle-play?f=classic&s=solid https://fontawesome.com/icons/trophy?f=classic&s=solid https://fontawesome.com/icons/list?f=classic&s=solid https://fontawesome.com/icons/circle-pause?f=classic&s=solid https://fontawesome.com/icons/refresh?f=classic&s=solid https://fontawesome.com/icons/volume-xmark?f=classic&s=solid https://fontawesome.com/icons/volume-high?f=classic&s=solid	Activity_main.xml Game_view.xml Scoreboard.xml	Font Awesome Free License

FreePik Icons:  	https://www.freepik.com/icon/vibrating_4949090#fromView=search&page=1&position=20&uuid=4a4974cc-16f3-421c-9277-051c0b12b2d8 https://www.freepik.com/icon/vibrating_4949088#fromView=resource_detail&position=1	Game_View.xml	FreePik Content License
FreePik Images:    	https://www.freepik.com/free-vector/hand-drawn-container-ship_20082216.htm#fromView=search&page=1&position=24&uuid=743a57a9-111b-4c69-99d7-4f5552839ab8&query=cargo+ship+icon https://www.freepik.com/free-vector/hand-drawn-importation-concept_20123511.htm#fromView=search&page=4&position=8&uuid=d49d5753-dec5-4256-abc3-c7988acf30e9&query=Container https://www.freepik.com/free-vector/cargo-ship-with-colorful-containers_416324	App_launcher Icon Main_activty.xml Game_view.xml Map_view.xml	FreePik Content License

	349.htm#fromView=keyword&page=1&position=3&uuid=9e438a77-8233-473d-a4b9-e6ac5d25534e&query=Freight+ship+no+background https://www.freepik.com/free-vector/top-view-world-map-background 18921804.htm#fromView=keyword&page=1&position=0&uuid=2b65836b-58c6-48b8-98b6-fe72a6d89bad&query=World+map+cutout+geology		
Pixabay: Wave Audio	https://pixabay.com/sound-effects/waves-white-noise-9777/	Game_view.xml	Pixabay Content License