

Take a Leap: A collaborative platform game

Anna Weiss

Chalmers University of Technology
Gothenburg, Sweden
annawei@student.chalmers.se

Alexander Yeh

Chalmers University of Technology
Gothenburg, Sweden
alex@yeh.nu

Simon Nilsson

Chalmers University of Technology
Gothenburg, Sweden
simon@simphax.com

Marcus Tyren

Chalmers University of Technology
Gothenburg, Sweden
marcus.tyren@gmail.com

ABSTRACT

Take A Leap is a collaborative 2D platform game that requires smartphone screens to be joined together to reveal increasingly more of the game world and prevent the character from falling. Users work together to keep the character alive by aligning their devices according to what platform appears on their device at a given time. This project is developed in Swift for iOS and utilizes a framework for joining the screens.

Author Keywords

Mobile app; mobile game; platform game; collocated interaction; iOS game; collaborative game; mobile bubble

INTRODUCTION

As mobile devices have become ubiquitous permeating all aspects of private and professional life in the matter of a few years, the extent to which this has affected society is still largely uncharted territory. How the use of this proverbial “Swiss Army Knife” [6] will continue to expand the associated socio-cultural implications are difficult to predict. Gradually emerging is the tendency of people to rely on their devices in all sorts of circumstances, that are not limited to obtaining information and communication on the go to simplify life, but extend to education, productivity, and entertainment. Some of the supposedly helpful applications however, have evolved into time-consuming distractions that actually present an obstruction for managing everyday tasks. Part of this development is the pervasive scene of people habitually interacting with their device, eyes glued to the screen, instead of interacting with their environment and peers. This has been termed the “Mobile Bubble” by Lundgren and Torgersson [3].

Currently, there are several movements both in research and the industry to counteract this development. Apps like “PauseAble”¹ or “Offtime”² address this issue by encouraging people to not use their smartphones while Håkansson is dealing with the question of how to redesign the time-stealing apps themselves by researching the merits of “designing for enough” [1]. Yet another initiative to get people to look up from their devices is “Bursting the Mobile Bubble” [4], which actually utilizes these very devices and turn them into tools

that encourage its users to interact with their current physical environment and the people in it.

Take A Leap is a collaborative platform game that aims to burst one’s individual mobile bubble, and encourages people to have a fun experience outside of their personal smartphone screens. It creates an instant, mobile playground that is accessible to a group of collocated people. Instead of making players compete against each other, the concept centers around making players work together and coordinate their efforts to reach a common goal.

RELATED WORK

In this section, influences of Take A Leap will be discussed, presenting a selection of other projects that have addressed various angles relevant for this game. These include other platform games that involve the automatic movement of the game character, and also other mobile device applications that require more than one device simultaneously. The idea of combining screens together is of course not new, there have been numerous undertakings involving this concept. Some are productivity apps, and they use different approaches and technologies to join screens.

Bit.Trip Runner

Bt.Trip Runner³ is a 2D platformer. The game is split into different worlds, each consisting of eleven stages and a boss at the end. The player cannot control the movement of the character from left to right, as this occurs automatically, instead he controls the jumping and sliding of the character in order to evade obstacles and collect points. If the player is hit by an object, the game is reset to the beginning of the level, initiating the running again automatically.

iBrainstorm

iBrainstorm⁴ uses a bluetooth connection between a central iPad and up to four iPhones. The idea is that the main area - the iPad - is used as a canvas for gathering all the “Post-it notes” that each iPhone provides by flicking them towards the iPad. It is aimed at replacing traditional brainstorming methods that use pen and paper from analogue to digital. While

¹PauseAble - Relaxation at Your Fingertip: <http://www.pauseable.com/>

²(OFFTIME) - Focus Digital Balance: <http://offtime.co/>

³BIT.TRIP RUNNER: <https://totallychoice.com/games/bittrip-runner>

⁴“Think, Share Repeat.” iBrainstorm: <http://www.ibrainstormapp.com/>

it requires several devices, the joining of screens occurs on a more abstract level, than with Take A Leap, since the game requires the connection to be visually and temporally seamless so that a continuous experience is guaranteed. The synchronization of screens in an iBrainstorm session is not as crucial to the application's success.

Mosaic SDK

Mosaic⁵ is an open source project to adjoin mobile devices by swiping across from one screen to another. By having a screen in the center displaying an image, other screens can join in on that picture by swiping over the surface in the direction of the target screen. In that way the picture resizes to account for the additional space the extra screen provides, and thus makes a bigger picture.

STORY

As the premise of the game the following story was conceived:

The legend of Thlen is an ancient story of an alien catball who lived in space. She was traveling through galaxies and nebulas, in search of friends but never found any sign of another creature to play with. Growing lonelier and lonelier, one day she came upon a distant system of planets on the outskirts of a small galaxy. There she spotted a little planet that seemed like none of all the others she had ever encountered. Her curiosity tickled, she decided to take a closer look. As she entered the atmosphere she discovered more and more of the planet's uniqueness, noticing a great variety of fascinating landscapes and - as she drew closer - exotic matter in all kinds of colors - mostly green - that seemed to be growing out of the ground. What was this magical place?

The legend then continues that Thlen fell in love with this planet's unique appearance and that she was most fascinated by the diversity that she found in the oceans, swimming underwater and this is also where she finally found friends: A group of fish, and they had many adventures together.

CONCEPT AND GAMEPLAY

The idea for the game was based on the mutual conviction that a collocated game would be ideal to foster collaboration among all members of the game effort. The desire to create an application that encourages users to create together, or to reach some sort of common goal by joining forces, so as to create an experience that brings participants together on some sort of conceptual niveau too - not merely spatially - instead of fostering adversity.

The premise of Take a Leap is to help the character Thlen finding a friend. Thlen moves automatically at a set pace, running on the platforms that are present in the world and jumping off of them to another. Users have no control over Thlen's movement but instead have to try and "catch" it by aligning another phone with a new platform that it can land on. The game continues as long as Thlen does not fall off a platform, and it is up to the group of users to collaborate and make this go on for as long as possible. Each platform that is reached means a point for the group, so the longer the game

⁵Mosaic: <http://www.mosaic.io/>

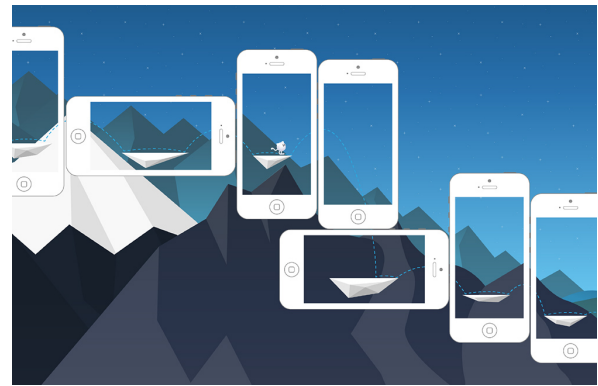


Figure 1. Concept sketch of the game progression

continues the more points are collected. This score is shown at the end of the game, and the players can then try to beat that high score in the next game.

IMPLEMENTATION

Gameplay development

Having players contribute to the gameplay by adding their own device as a game tile to play with makes the game appear intriguing from the first glance. Based on developing a simple game, a 2D platformer with jumping and running, and presenting it in this unusual form appeals to prospective users. Many are already familiar with how a platformer works, and Take A Leap offers a unique way of collaborating with friends to beat the highscore together. The reason behind making it flexible by not having a limit on devices or participants was to involve more players, resulting in a varying game experience depending on how many players or devices are present. This could either make it easier or more hectic and it is up to the players to figure out what works best for them.

Design

To create an immersive experience, the game is set in a richly illustrated world. The illustration style is inspired by games like Monument Valley and Alto's Journey, where the visuals manage to create a magical atmosphere that makes playing all the more enjoyable. The first step was to create a concept for a world, and since the general movement of the jumping character is generally to the right and down, it was natural to adopt a setting starting really high up in space and then moving down gradually touching planet earth; Beginning with the peaks of icy mountains, progressing to rocky mountains then turning into woods and hills further into down the valley, continuing into fields, more hills and more tropical vegetation, merging into sandy dunes until reaching a beach winding along the seaside. Then Thlen continues her journey further underwater, ending up on the bottom of the ocean floor, where she meets a fish and they become friends, which is the final goal of the game. All graphics are conceived specially for the game and built as vector graphics in Adobe Illustrator. As a creature to inhabit this magical world, the character was derived from the Legend of Thlen, and its appearance developed to appeal to the players' protective instinct. Then was

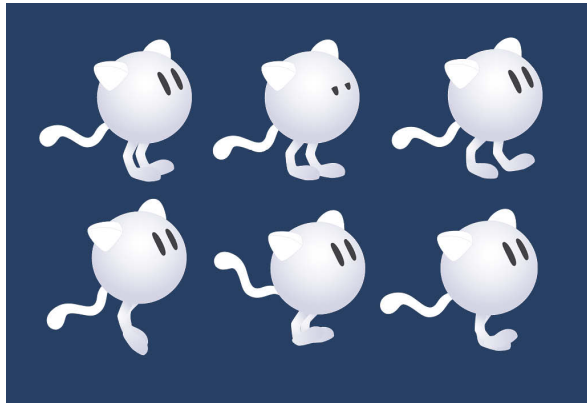


Figure 2. Different states of the character

first sketched on paper, and then digitized in Adobe Illustrator. The resulting cuteness-exuding catball alien has a curious and playful personality, embodied in its movements, which have been animated in Adobe After Effects and exported as png sequences to be fetched and displayed at different game states.

The landing platforms that catch Thlen and are distributed across the player's screens appear as floating icebergs in pointy triangular shapes to stand out from the more organically shaped background. This is all concerning the immediate game experience, however there are also the screens preceding and superseding the actual play. To set up the game, several menu views were developed and designed to fit into the general look and feel. Especial focus was given of how to explain to the users what to do. How can the desired actions be communicated in the most efficient and unobtrusive way? Through descriptive text, or through images or diagrams? To clarify how to place smartphones next to each other and use the pinch motion we opted for a quick diagrammatic animation that is played before each game instance, which was also created in After Effects. Also, a pause screen (that also appears in case of lost network connection), game over screen, and high score screen were conceived.

Pinch framework

Pinch is an interface connecting the displays of multiple touch devices together. It was developed by a research group at the Tokyo University of Technology in 2012 [5]. When the user places the thumb and index finger on two adjacent screens and executes a pinch motion, the screens "up". The screens can be placed in any way, vertically or horizontally, and the pictures can be matched up with the "pinch-close" gesture, if they are misaligned. Originally, the connected devices share each other's position and screen size via Wi-Fi. With this system, it is possible to connect devices with differently-sized screens, such as iPhones together with iPads. The code was updated from Objective C to make it compatible with Swift. Furthermore, a change was made in Apple's libraries to optimize the synchronisation of phones so that always the currently best working connection is chosen, whether it is Wi-Fi or bluetooth.

Synchronizing character and background positions

The Pinch library was utilized for the game to get the relative positions of each phone that is joined in the game. The users will use the "pinch-close" gesture to connect the phone screens together to create a big canvas for the game. The relative position retrieved from the library is used to determine the position of the background image for each phone and also to give the character the correct trajectory when it jumps from one phone to another.

When the character reaches the end of a platform she will perform a jump that makes her fly off the visible screen space. To have the character jump between the phones, the game detects when she is outside the screen space and examines the joined phone's positions provided by the Pinch library. It determines which device is closest to the character's position and sends the character's position and velocity vector to that device. The character is then removed from the phone she is leaving and then created on the other phone on the provided position converted into the new coordinate system, again using the Pinch library. The provided velocity vector is applied to the character so she will continue on the same trajectory that was commenced on the initial device.

The background is calculated to be relative to the device that currently owns the character. The calculations in the library assists in converting points in a coordinate system of one device to another, as well as providing the angle between them. When a device has been handed over the character, it will broadcast to the other phones that it is now the new reference point of the background and which offset is now set for the background position. Any new device that joins into the game canvas will now use its position and its offset as the reference.

Physics

By using SpriteKit's ⁶ physics engine, physics bodies were easily added to the scene; one for the character and one for the platform, and simulation would be performed on them. All physics bodies have their own properties to define how they are simulated. These could be mass, velocity or how the object is affected by gravity. For this project, only a few of these properties were interesting, such as the character being affected by gravity but not the platform, and making them collide with each other thus. Every time a scene computes a new frame of animation, it simulates the effects of forces and collisions on physics bodies connected to the node tree. Inducing continuous movement on the character by having her velocity in the x-direction being constant was also used. To make the character jump automatically when reaching the end of a platform, an impulse was induced to temporarily add velocity in the y-direction. To detect when a contact between character and platform has occurred, and when its contact is lost, `collisionBitMasks` and `contactTestBitMasks` is used together with `SKPhysicsContactDelegate` protocol and methods `didBeginContact` and `didEndContact`. Using only a `characterCategory` and a `platformCategory` to match them

⁶"SpriteKit Programming Guide." About SpriteKit: https://developer.apple.com/library/ios/documentation/GraphicsAnimation/Conceptual/SpriteKit_PG/Introduction/Introduction.html

against each other, but can be extended in the future for adding obstacles or collectables. These methods were also later used to detect when to apply certain animations, such as jumping or running animations.

Animations

Thlen has five different animation states: Standing, idle, running, jumping and landing. For instance, if Thlen is about to jump, the jump animations will be displayed, and when Thlen enters a new state a new animation will be displayed. It uses pre-existing functions that was implemented in the physics section to detect if Thlen is standing, idle, jumping, landing or running. As an example, when Thlen has landed and is moving horizontally to the right, the running animation is displayed.

Communication

The game is largely dependent on the synchronization of different values, like background position, player position, score count etc. The Pinch library utilizes a library from apple called Multipeer Connectivity. For convenience and consistency, the same connection was used for the game specific communications. The Multipeer Connectivity library has the ability to automatically discover and connect to devices in the local area based on a common service id. The library will automatically connect using the best way of communication. Depending on what is available it will use Wi-Fi, Wi-Fi Direct or Bluetooth personal area connections⁷.

Playtesting

To determine the game's entertainment factor a playtesting session was conducted. It included four users and four devices. First, was a test to see, whether the animations for how to do the swiping for connecting the devices was intuitive, and make sure the gameplay was not cumbersome. Secondly, the right speed for running and jumping needed to be tested, to achieve a pace neither too slow nor too fast and thereby diminishing the experience. The feedback was generally very positive. Animation and swiping was indeed intuitive and everyone enjoyed the basic idea of the game. Some wanted the speed to be more hectic, whilst others wanted to enjoy it at slow tempo with less stress. Some suggestions were to add the function to change difficulty, i.e the speed, or heighten the speed as the game progresses. At the time of playtesting, there was no option to restart the game immediately upon game over, but one had to reset the whole app, which was the source of some complaint. Another suggestion regarding adding collectables, was to place these on the platform as coins, to add an incentive to try and land early on the platform to get more coins, thus creating another aim for the players to achieve.

RESULTS

The resulting product is a collaborative 2D platform game. The game consists of a character moving from left to right and jumping off platforms to reach lower ground. Users play the game together by aligning phones so that the character lands on a new platform and can continue going forward.

⁷Multipeer Connectivity Framework Reference: <https://developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MultipeerConnectivityFramework/>

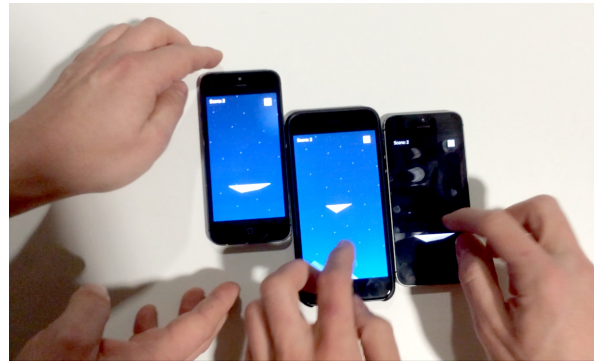


Figure 3. Game in action

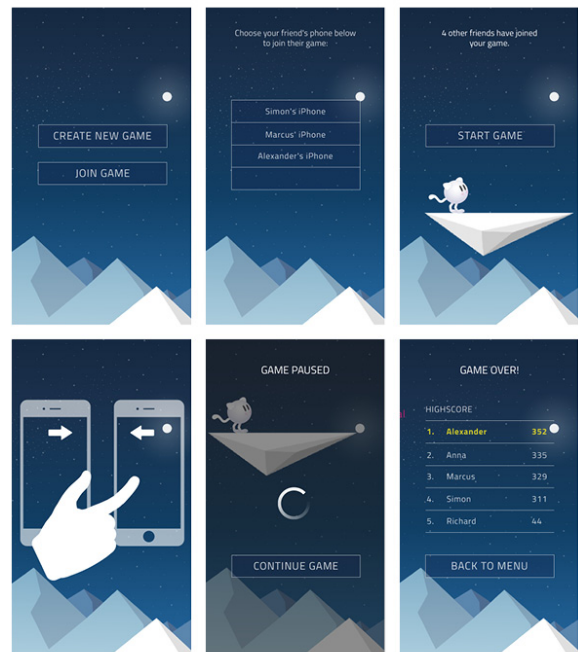


Figure 4. Menu screen mockups

Before it jumps over to the next screen, the users have to "the smartphones with a pinch motion for the level to extend horizontally and vertically. Users get one point for every successful jump between two phones if they also manage to land the Thlen on a platform. If they fail to "catch" the Thlen, i.e it misses a platform, the game session ends and the final score is shown. It works for any number of players and devices, the minimum requirement being two devices.

FUTURE WORK

Over the course of the project many features were thought of to enrich the gameplay and extend the game. However, due to time constraints it was necessary to focus on the core functionalities, therefore a selection of proposed additions in the future is introduced below.

To make the game more random and challenging, Thlen could change the horizontal direction upon landing on a platform, either continuing to the right, or changing to the left. In that case the game will feel much more different every time it is played it since the phones have to be aligned both to the right and left at random.

After some time, the game may feel a little repetitive as it only includes the character, platform, and moving background. Adding coins or other kinds of collectible items could add more variety to the game and thus make it more enjoyable. This could be further enhanced by introducing powerups or obstacles. To add more suspense to the game, and creating a sort of obstacle mentioned before, adding empty screens, just background and no platform is easily implementable. That way those screens will work as “mines” as the users would have to collaborate to resort the devices in order to help Thlen survive the jump.

In an attempt to give users more control over the forces in the game, implementing some functionality that affects Thlen’s path of travel is a declared goal for the future. The first idea was the “tilt to change trajectory”, which means giving users the power to tilt their phone to shorten the trajectory of Thlen’s jump. Furthermore, to allow users a chance for a breather in the very stressful gameplay environment, a timeout for a few seconds was also debated. By shaking the phone Thlen would fall over and stop before commencing its movement again.

The Pinch library provided to us was a great assets and worked almost flawlessly, however sometimes the connection between the devices would drop. In order to remedy that, the user would have to shake or rotate the device in order to reconnect.

Also included in the future work is creating more levels, which are needed to increase the re-playability. Generating novel levels and perhaps add achievements tied to each level that can be added to the points, jumps or players involved to complete the level.

CONCLUSION

Given the responses from the participants of the playtest sessions, Take A Leap appears to have met the underlying goal of making people interact with each other, instead of with just their respective smartphones. Also, from observing the playtesters it became clear that this game achieves the goal of instantly constructing the conceptual playground that was mentioned in the introduction. Like a boardgame it succeeds in creating that “Magic Circle” [2] among the players. Take A Leap became unexpectedly physical as the playtest participants were reshuffling the devices - and themselves - to align the layout in the game and therefore bump into each other by mistake. This introduced another layer of interaction, on top of the verbal communication to organize the collaborative effort.

In terms of the work process, the Pinch library was much appreciated and reduced the workload considerably, encountering only very minor issues. On another note, the Multipeer Connectivity framework came as a welcome surprise as it was expected that the Pinch library would be using the same

network the devices were connected to. This allowed the devices to connect to each other regardless whether they were on the same network or not and was also able to connect via bluetooth, without the need to intervene from the developer.

In the end we would have wanted to add more features and more variation in the gameplay, hopefully saving it for future development.

ACKNOWLEDGMENTS

We would like to thank professor Takashi Ohta from Tokyo University of Technology for providing us with the Pinch library and James Wen for the supervision of our project.

REFERENCES

1. Maria Håkansson and Phoebe Sengers. 2014. No easy compromise: sustainability and the dilemmas and dynamics of change. In *Proceedings of the 2014 conference on Designing interactive systems*. ACM, 1025–1034.
2. Johan Huizinga. 1955. *Homo Ludens: A Study of the Play-element in Culture*. Beacon Press.
3. Sus Lundgren, Joel E Fischer, Stuart Reeves, and Olof Torgersson. 2015. Designing mobile experiences for collocated interaction. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 496–507.
4. Sus Lundgren and Olof Torgersson. 2013. Bursting the mobile bubble. In *First International Workshop on Designing Mobile Face-to-Face Group Interactions, European Conference on Computer Supported Cooperative Work, ECSCW*, Vol. 2013.
5. Takashi Ohta and Jun Tanaka. 2012. Pinch: an interface that relates applications on multiple touch-screen by ‘pinching’ gesture. In *Advances in Computer Entertainment*. Springer, 320–335.
6. Mahadev Satyanarayanan. 2005. Swiss army knife or wallet? *IEEE Pervasive Computing* 2 (2005), 2–3.