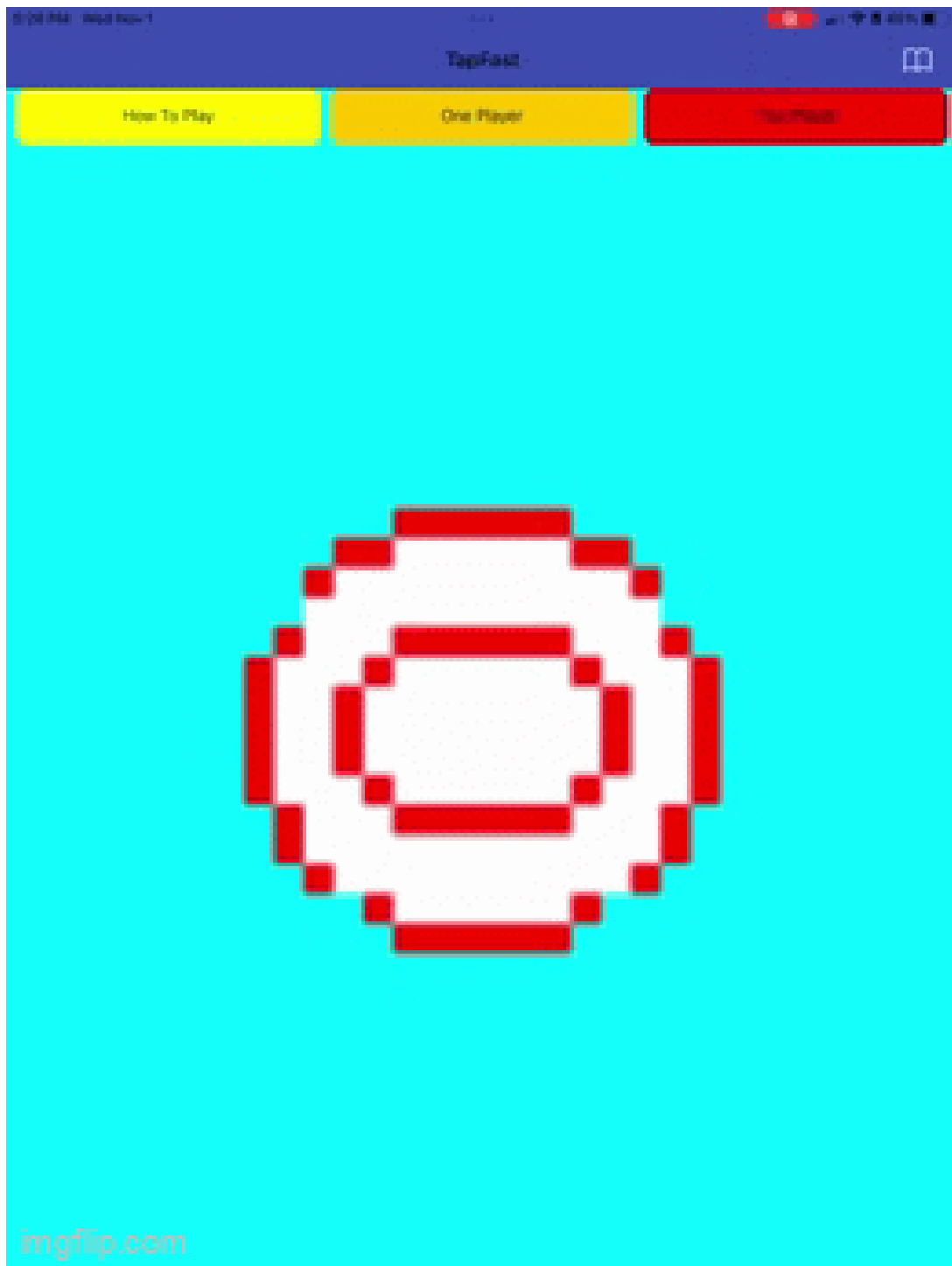


TapFast



Designers:

Ashton, Ambareesh

PLTW | Computer Science

Table of Contents

Table of Contents	2
Plan & Design Your Solution: SCRUM	3
Product Backlog	3
Sprint Planning & Sprint Backlog	3
Sprint 1	3
Sprint 1 Tasks	3
Sprint Retrospective (Individual)	3
Sprint Review (Individual)	3
Code File Links	4
Sprint 2	4
Sprint 2 Tasks	4
Sprint Retrospective (Individual)	4
Sprint Review (Individual)	4
Code File Links	4
Sprint 3	4
Sprint 3 Tasks	4
Sprint Retrospective (Individual)	4
Sprint Review (Individual)	4
Code File Links	4
Final Sprint Review & Retrospective (Individual)	4
Final Product & Justifications	5
Self Evaluation & Final Project Code	5
Interpretive Performance Guide	5
Program Code	5
Conclusion	5
_____’s Reflection:	6
_____’s Reflection:	6
Project Log & Brainstorming	7

Plan & Design Your Solution: SCRUM

Product Backlog

What big picture product features need to be completed in order to have a successful product. Note, in this section you might need to include sketches to better aid in a strangers understanding of the goal of this product.

Sprint Planning & Sprint Backlog

What are all the smaller tasks that need to be completed in order to complete your project, and how long do you think each task will take to complete (breakdown using hours or minutes, also use strikethrough text styling once a task has been completed...if it took more time add a **+ units** and if it took less time add **- units** you must use the highlight color).

Note: Make sure you add all the tasks required for the entire project in the table below.

[Insert rows as needed in the table below.](#)

Table 1: Main Table: Contain all the tasks

Tasks	Estimated Time	+/- Time	Completed By Date and Person name	Sprint #
Create animation	10 minutes	+10 minutes	Ashton Oct 29	1
Write 1 Player Code and How To Play Code	30 minutes	+10 minutes	Ashton and Ambareesh Oct 30	2
Write 2 Player Code	20 minutes	+20 minutes	Ambareesh Oct 31	3

Sprint 1

Sprint 1 Tasks

Copy and paste tasks, with time details from the sprint backlog and place them here.

Note: This will include the task you plan to complete in the first week from the above table.

[Insert rows as needed in the table below.](#)

Tasks	Estimated Time	+/- Time	Completed By
Create animation	10	+10 minutes	Ashton Oct 29

Sprint Retrospective (Individual): Ashton

What challenges as a team did you face that are NOT code related? Were you able to overcome those challenges? What is one goal you have to work better as a team (that is not code related)?

We struggled with the user interface of Piskel, though it was relatively intuitive after we learned it. We need to work better as a team by sharing the project, which is more efficient then one person doing and one person watching.

Sprint Review (Individual): Ashton

What challenges as a team did you face that are code related? Were you able to overcome those challenges? Do we need to make any modifications to our sprint backlog (add more tasks, put back tasks we did not finish, edit original tasks based on feedback, etc.)?

It was difficult for us to port the animation over to the MIT app inventor in a way that allowed the animation to play. In our first two attempts, we would transport it over to the MIT app inventor as an image, which defeated the purpose of the animation. We were able to overcome this.

Sprint Retrospective (Individual): Ambareesh

What challenges as a team did you face that are NOT code related? Were you able to overcome those challenges? What is one goal you have to work better as a team (that is not code related)?

Sending the MIT app inventor over the aia link was difficult, but we were able to overcome this after consulting a nearby group.

Sprint Review (Individual): Ambareesh

What challenges as a team did you face that are code related? Were you able to overcome those challenges? Do we need to make any modifications to our sprint backlog (add more tasks, put back tasks we did not finish, edit original tasks based on feedback, etc.)?

We did not face many code-related challenges. We needed to modify our sprint backlog by increasing the time it took because we were unable to finish it on the first day and therefore were forced to continue it at home.

Code File Links

https://cdn.discordapp.com/attachments/1169390887894253618/1169416216708919347/New_Piskel_1.gif?ex=655552b8&is=6542ddb8&hm=7e70aca1c6e3164f2648adebe05f527d025ad5c350587da1b3d3220a2e87f009&https://cdn.discordapp.com/attachments/1169390887894253618/1169416217325477949/target_1.gif?ex=655552b8&is=6542ddb8&hm=fb3218ecef2c94bd0eaa93168511a31a1f262ae7300bc17e02be085a8f9bfe91&https://media.discordapp.net/attachments/1169390887894253618/1169416217824595979/New_Piskel.gif?ex=655552b9&is=6542ddb9&hm=5fe65c65fbf3d6ee3e8f990a54df313cd4b5fe32a8da1cdb2cbbbeac326b417e7&=&width=28&height=28

Move onto the next sprint and repeat the previous steps to develop and test each feature until you have completed your project.

Sprint 2

Sprint 2 Tasks

Copy and paste tasks, with time details from the sprint backlog and place them here.

Note: This will include the task you plan to complete in the first week from the above table.

[Insert rows as needed in the table below](#)

Tasks	Estimated Time	+/- Time	Completed By
Write Player 1 Code	30 min	+10 min	Ashton and Ambareesh Oct 30

Sprint Retrospective (Individual): Ashton

What challenges as a team did you face that are NOT code related? Were you able to overcome those challenges? What is one goal you have to work better as a team (that is not code related)?

It was difficult for us to run it on android because the gifs were not loading despite its functionality when we were at home. We were deciding whether it was a wifi error or a code error. This could have been resolved if we had shared the code earlier so both people could test it, allowing less room for ambiguity with respect to the cause of the problem.

Sprint Review (Individual): Ashton

What challenges as a team did you face that are code related? Were you able to overcome those challenges? Do we need to make any modifications to our sprint backlog (add more tasks, put back tasks we did not finish, edit original tasks based on feedback, etc.)?

We struggled to make the timer go down by 1 second every time even though we set the increment to 1. We realized that our loop was running twice because of a coding error and we quickly fixed it. We did not need to modify our sprint backlog as things went mostly smoothly, and the source of the errors were quickly traced.

Sprint Retrospective (Individual): Ambareesh

What challenges as a team did you face that are NOT code related? Were you able to overcome those challenges? What is one goal you have to work better as a team (that is not code related)?

I accidentally closed it before everything was saved which ended up deleting some of our code on MIT App Inventor, forcing us to restart. Luckily I was able to get everything back really quickly since I remembered what we did the first time.

Sprint Review (Individual): Ambareesh

What challenges as a team did you face that are code related? Were you able to overcome those challenges? Do we need to make any modifications to our sprint backlog (add more tasks, put back tasks we did not finish, edit original tasks based on feedback, etc.)?

There were too many elements and it was easy to get confused by the component names. In particular, we kept making the wrong horizontal arrangement invisible and we believed it was not loading for a long time until we finally realized what was wrong the day after at school.

Code File Links

<https://drive.google.com/file/d/1uogai34l6yeJqme97Pe9MKxdnegeQHtl/view?usp=sharing>

Move onto the next sprint and repeat the previous steps to develop and test each feature until you have completed your project.

Sprint 3

Sprint 3 Tasks

Copy and paste tasks, with time details from the sprint backlog and place them here.

Note: This will include the task you plan to complete in the first week from the above table.

[Insert rows as needed in the table below](#)

Tasks	Estimated Time	+/- Time	Completed By
Two Player Mode	20	+20	Ambareesh Oct 31

Sprint Retrospective (Individual): Ashton

What challenges as a team did you face that are NOT code related? Were you able to overcome those challenges? What is one goal you have to work better as a team (that is not code related)?

Our loop was not running as intended because we would get an error whenever we tapped on the first circle. We were able to trace the error by printing a word at every step of the way so that we could systematically narrow down the possible source of error down to the loop, where we discovered that it was not being stopped by the timer.

Sprint Review (Individual): Ashton

What challenges as a team did you face that are code related? Were you able to overcome those challenges? Do we need to make any modifications to our sprint backlog (add more tasks, put back tasks we did not finish, edit original tasks based on feedback, etc.)?

We got into many arguments as to the source of our problems that wasted a lot of time. This added time needed but did not affect our overall result.

Sprint Retrospective (Individual): Ambareesh

What challenges as a team did you face that are NOT code related? Were you able to overcome those challenges? What is one goal you have to work better as a team (that is not code related)?

Our app kept crashing because one of our loops did not end even when the time expired. We were able to overcome this with relative ease because we encountered this problem before.

Sprint Review (Individual): Ambareesh

What challenges as a team did you face that are code related? Were you able to overcome those challenges? Do we need to make any modifications to our sprint backlog (add more tasks, put back tasks we did not finish, edit original tasks based on feedback, etc.)?

Our piskel would mysteriously disappear and we were unsure whether this was, once again, a bug with the code or the wifi. We tried changing the canvas size to make sure that it should be showing on screen but it would still disappear. We concluded that it was, in fact, a wifi issue.

Code File Links

<https://drive.google.com/file/d/1uogai34l6yeJqme97Pe9MKxdnegeQHtl/view?usp=sharing>

Note: Add more sprints if needed until you complete the project, or the project is due!

Final Sprint Review & Retrospective (Individual): Ashton

Type Here...***Organize and Format Nicely!!!***

We were able to design the app through a relatively simple process. I would have liked the app to have included more features, however. I believe that our game design may have been too simple, and in retrospect I wish we had spent more time deciding the direction of our project before actually starting the code. I believe that we were too focused on completing the assignment and not enough on making an app that fulfilled our creative passion at the end of the day. One major milestone we were unable to reach was a two-player mode where players could alter the course of the game by tapping on different items. However, we did have the time to implement all of this because the code became too convoluted and the elements were so high in number that it was nearly impossible to keep track of everything. When we tried putting everything together, there were so many mistakes in our code that we just reverted back to the original iteration. We definitely would have spent more time working in class rather than on unproductive arguments if we redid this project. We should have also shared it with one another on the first day and given contact information, as it was very difficult to reach conclusions while we could not communicate at home.

What went well through this process, what did not go well. Make note of any features you did not have time to develop and any milestones you did not reach. Also reflect on what you would have done differently if you got to repeat this entire project/process all over again

Final Sprint Review & Retrospective (Individual): Ambareesh

Type Here...***Organize and Format Nicely!!!***

There were a lot of coding errors that we encountered, such as the piskel not loading correctly. This was a major issue as we could not determine whether this was caused by the wifi or not. If I were to redo this again, I definitely would spend more time testing it at home so that we could be more certain about whether or not it was a wifi issue at school that was causing the app to not run. One milestone we did not reach was a more diversified game-mode selection screen. We could only make the one player and multiplayer modes, because we could not agree on the degree of difficulty and more intricate features about the other game modes. We also did not make our original app that we agreed on because we thought it was not original. In general, there were too many changes that happened during the creation process that constantly forced us to restart. A general lack of communication and collaborative working while we were at home also made the teamwork aspect very difficult.

What went well through this process, what did not go well. Make note of any features you did not have time to develop and any milestones you did not reach. Also reflect on what you would have done differently if you got to repeat this entire project/process all over again

Final Product & Justifications

What extracurricular/club activity does your product incorporate, and why did you and your partner choose to work on this?

Insert a Video of your final app in action here

<https://drive.google.com/file/d/1nyslyFzN6gPw-1PbGAXK977jXO0rRgTt/view?usp=sharing>

Final .aia file link:

<https://drive.google.com/file/d/1uogai34l6yeJqme97Pe9MKxdnegeQHtl/view?usp=sharing>

Final .apk file link:

<https://drive.google.com/file/d/1c1CfN-N994Pqf5k7WkkkF7KtjowyuYx/view?usp=sharing>

Explain your App and how it works as well as what the meaning behind it is. What does it represent and how does it tie to your extracurricular/club activity?

We have both done competitive math and are preparing for the SAT, but a major issue in both of these is the complete inability to concentrate under time pressure. Therefore, we set a strict time limit combined with a competitive element that emulated some of the key components in these activities. We hope that, despite the simplistic nature of this app in comparison to these tests, we are able to derive some skills from the teamwork and perseverance that went into this project.

Self Evaluation & Final Project Code

Interpretive Performance Guide

To help prepare you for the next Computer Science Course (AP CSP) I want you to look at and evaluate what score you would get based on the following [Interpretive Performance Guide \(Click Here\)](#). Note: if you do not satisfy and show proof in this documentation of all items in the Low category give yourself a “o” for that row.

Based on the Guide My partner and I would get: 17/18

Justification for that score, go into detail on what rows you think you would get and why!

Row 1 (Collaborating on Computing): $\frac{2}{3}$. This is because we were generally productive but were unable to work productively in class. However, we were able to resolve many communication issues at home and worked together at home, which alleviated many of the teamwork problems. Our collaboration ultimately impacted the final design because we were unable to reach an agreement over many of the design elements for the different game modes, leading to an unfinished project. However,

we were able to resolve these differences towards the end and, if given more time, I genuinely believe that we can come to an agreement to create a better project.

Row 2 (Developing a Program with A Purpose): 3/3. We were able to run the program in a working manner. Although the gifs could not load, this was certainly not a problem with our code because it was able to work at inconsistent intervals. We believe that this is relevant to MIT App Inventor or our wifi, because we were able to get it to work sometimes. The development of this app had a clear goal in mind. We had difficulty coming up with what we wanted our final product to be at the start, but once we had a shared vision for the end product we were able to get there without changing our mind. We got frequent bouts of inspiration along the way, but ultimately stuck to our original understanding of what the app would become.

Row 3 (Applying Algorithms): 3/3. One algorithm we used was the timer. This was particularly tricky due to the aforementioned problematic nature of the varying components. We had to make sure the visibility and functionality of each of the non-pertinent components was turned off, else it interfered with the functionality during the desired interval of time. Here we combined what we learned from all the prior lessons, concatenating a changing global variable into a printed string using both arithmetic and logical operators.

Row 4 (Applying Abstractions): 3/3. We were able to effectively develop an abstraction using pseudocode before starting. This was divided into the initial screen, with 3 buttons (one for the rules, one for single player, and one for multiplayer), and the functionality of each of the buttons. The rules would display a message that would go away after another button was clicked. The single player mode has a target that moves around the screen that serves as practice for the multiplayer variant, where targets of different appearances move around the screen and each player needs to tap their corresponding target. Both of these modes include a timer and a score for each respective player that both have an increment of 1 to avoid making the User Story overly complicated. One procedure we used throughout the course of this code was generating a random place for it to appear. Here we integrated what we learned from the interactive lesson where Mr. Baez showed us how to center the image based on the image's proportions. We generated a random number between the two endpoints and set it for both the width and height while running it in a loop, all the while recording the score for each player using a global variable. The parameters for this procedure were the length and height of the screen, which we could adjust using a built-in scale factor that allowed us to forgo using a numerical system that would vary from screen to screen. That way, the image should always appear on screen within the scope of a canvas that spans the entire screen. The app would have been less organized without the abstraction and, even if the final product may not have been affected apart from the interface design, would nonetheless have taken significantly more time to program without a concrete goal in mind.

Row 5 (Testing and Debugging): 3/3 We were able to debug the infinite loop issue with the app crashing the first time the symbol was clicked by connecting a problem we had with a previous project with this one. The issue with a loop pertaining to the timer was also fixed when we printed a value each time the loop ran, indicating the loop was running twice per second, which is double the intended amount. This allowed us to trace the source of the error to the root cause almost immediately. One area where the user interface was redesigned was when we had multiple buttons on the initial screen, one of which was a menu that showed the different game modes. However, we realized that it would make more sense for the user if the buttons were located on the home screen, even if the original user interface was more organized for the programmers. This connects to user bias

because most gamemode options are located on the home screen rather than being accessible through several buttons because it is the most fundamental element of a game.

Row 6 (Communicating Around Computing): We were able to communicate effectively by sharing contact information after the second day of coding so we could communicate at home. This allowed us to test the app under more circumstances, allowing the probability of the app failing to function to diminish substantially.

Program Code

Insert as many screen captures as needed of your program code with readable comments for the major features and functional aspects of your App (Note: I suggest you use a table to help organize your images and comments (comments can use app inventor comments or text in this document)):

Conclusion

Ashton's Reflection:

- 1) My contribution to this project came in the form of making the piskels, making the general app design, writing pseudocode to organize thoughts, and coding the one-player portion. I first came up with the idea of making animated piskel targets that the player must tap on in order to gain points. I helped program the one player game-mode, which included a clock algorithm, a point algorithm, and a procedure within a loop that randomly changed the location of the piskel within the span of the canvas. I helped with the color scheme. However, most of my contributions were not code related. I initiated conversations through discord, and asked to exchange contact information at school so we could be more productive and coordinated at home. Because Ambareesh did most of the code, I compensated for this by doing more of the documentation.
- 2) a) The purpose of the program is particularly interesting because we shared the same vision but for different goals. We agreed that this is a program focused on competition between friends. However, I took inspiration specifically from apps that train your mouse speed and accuracy by making you click on targets really quickly, which helps chess players be more quick in hyperbullet chess. I figured that we can translate this app to a phone, which is another common place people play hyperbullet on.
b) One mathematical operator this program uses is the point and time system. It uses a for loop, which is effectively a recursive mathematical function, to dictate the start and end for a loop that adds points and subtracts time from a starting point. Mathematical operators also compare who got more points in order to determine a winner.
c) Our algorithms were primarily focused on calculating the points and times within the game. We used boolean expressions and chained conditional statements to develop both game modes, where points are taken away based on player input.
d) The abstractions we created were primarily a summary of our pseudocode in a comprehensive and organized manner. It was divided into the main screen, the single player,

and multiplayer section. This helped us in our code because it gave us a more organized vision and implanted an end goal that we could strive for. With these abstractions, we could avoid many pointless arguments in class over design philosophy that could waste time and prevent us from making use of work time in a productive manner.

Ambareesh's Reflection:

1. My contribution to this was to be a core developer. I did a little more than half of the code and worked with the majority of the bugs. I worked on the basics of the one player code, while Ashton cleaned it up a bit. I also did the 2 player and How To Play part of this app, although Ashton and I brainstormed the layout and the generic functionalities together. Ashton initially explained to me the design of these different modes and I then proceeded to code these 2 parts and debugged it, mostly myself.
2.
 - a. This is just a fun or a time-killing game which can be played during things like car rides, or when you and your friend are bored and are down to a competition. This game can be used for alternate purposes such as working on improving your reaction time, hand eye coordination, or just concentration in general.
 - b. This program uses logical and mathematical concepts at places such as finding who scored more points in the 2 player mode by comparing both the players' scores using comparison operators, which is the logical part of it. The mathematical part is when the seconds get subtracted by 1 as the game passes on. This can be seen in the One player mode or two player mode.
 - c. There are a lot of algorithms in this code such as the conditionals used. For example, **IF** the timer hits 0, **THEN**, the code should send the results and have a reset option available.
 - d. One place where I used abstraction is by hiding the main menu to switch between the game modes (one player mode to two player mode, or vise versa), which reduces the clutter on the screen making it less complex for the player and the code to function properly, as there is less components on the screen.

Answer the 1.1.6 conclusion question here...be DETAILED!

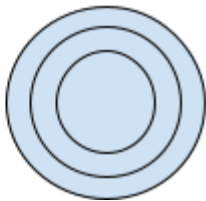
Project Log & Brainstorming

This table MUST be detailed! It will be how I grade a minimum of 10% of your score for this project. You must record an entry for every day of this project, even if we do not meet for class.

Brainstorming: Steps 1 through 7: Type Here...Organize and Format Nicely!!! You should also include brainstorming & developmental sketches here. They can be google drawings, photos of your hand sketches, etc. No matter what though, Organize and Format Nicely!!!

Reminder, your App must meet ALL requirements given in 1.1.6 PLTW instructions & incorporate some extracurricular/club activity today!

Score: ###Timer: ###



Timer

Player 1 Score:

Player 2 Score



Ashton	Ambareesh
<p>Oct 23: We mostly brainstormed today and came up with an app design that was original. We wanted to do either Wordle, Hangman, or some board game such as Tic-Tac-Toe.</p> <p>Oct 24: Mr Baez informed us that we would not be able to do Tic-Tac-Toe as it is unoriginal, so we decided on a different model of a game that involved fast tapping.</p> <p>Oct 25: Since today is a Wednesday, we were unable to meet and we did not have contact with each other. However, we bumped into each other at lunch and agreed on a two player mode. We disagreed on the interface but settled on two buttons that revealed the game modes and rules.</p> <p>Oct 26: We went on MIT App Inventor and designed the initial screen. There is a button that reveals the rules of the game and another button that leads to a selection screen where you pick the two modes. I also designed the piskel model, which Ambareesh helped port over the MIT App Inventor. However, it is still a static image of the piskel MIT App inventor, but we were unable to fix it today in time.</p> <p>Oct 27: We still have not exchanged contact information, which I plan to do on Monday or next time we meet. We did a bit of programming at home, including writing out the single player pseudocode and implementing the different icons as a tappable target.</p> <p>Oct 28: Ambareesh and I made the timer for the one and two player modes.</p> <p>Oct 29: I made the algorithm that determines the score of the player and the procedure that changes the location of the icon.</p> <p>Oct 30: I polished the start screen with a blue background and an animated target at the bottom (though it does not animate when loaded up on Android).</p> <p>Oct 31: We put the photos that we've taken along the way into our documentation.</p> <p>Nov 1: We finished the documentation.</p>	<p>Oct 23: We discussed what we wanted the app to look like and decided on either Wordle or Hangman.</p> <p>Oct 24: We changed the model of our plan from Wordle and hangman to a game where you compete with someone else to try and tap a target as quickly as possible.</p> <p>Oct 25: I thought we should make a two player mode that allowed both players to compete by tapping on icons that had different appearances. With this, we also figured out how we can design the initial screen.</p> <p>Oct 26: We made the home screen, but I believe it is too inconvenient and has too many buttons. I think we can change this towards the end, as it is an easy fix that does not require any debugging. We made the target design and set it as an image sprite in MIT App inventor. It was not working as intended, however, because it is not a gif like it is in Piskel. We are still trying to figure out how to fix it.</p> <p>Oct 27: I managed to turn the Piskel into a gif in MIT App Inventor, but while testing, it remains an image. I ran it a few more times and sometimes it would be animated. I believe that this is not a problem with our code and the problem lies with my wifi.</p> <p>Oct 28: I coded the timer today using an algorithm for the two player mode</p> <p>Oct 29: I made Ashton's algorithm for the one player applicable to the two player version.</p> <p>Oct 30: I changed the initial screen so that it no longer starts with 2 different buttons that lead to the gamemode and the rules, and instead put everything in the initial screen so that it is way easier for the user to navigate. We also exchanged contact information.</p> <p>Oct 31: We completed a large chunk of the documentation.</p> <p>Nov 1: I recorded a video of our app for the documentation.</p>

```

when Screen1.Initialize
do set InfoMainScreen.Visible to true

```

```

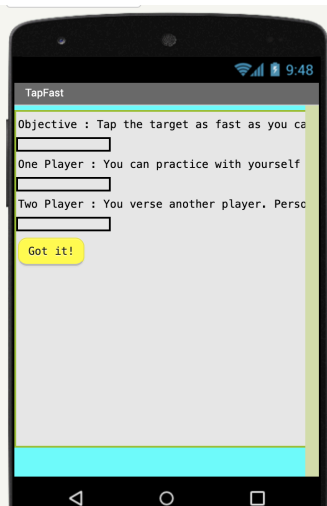
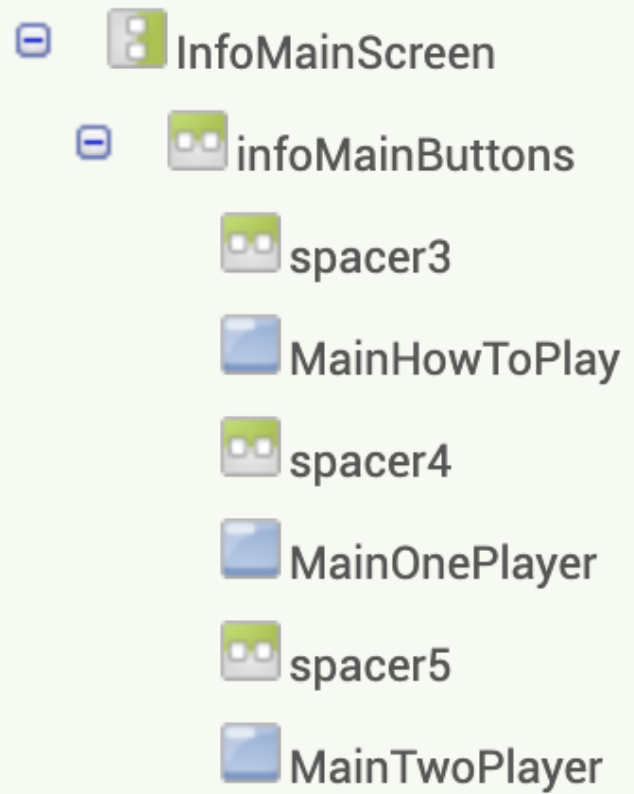
when MainHowToPlay.Click
do set InfoMainScreen.Visible to false
   set HTPall.Visible to true

```

```

when HTPgott.Click
do set HTPall.Visible to false
   set InfoMainScreen.Visible to true

```




```

initialize global OPscore to 0
initialize global OPTimer to 30

when MainOnePlayer .Click
do
  set OPRReset . Visible to false
  set OPTimerLabel . Text to "Timer : "
  set InfoMainScreen . Visible to false
  set OPall . Visible to true
  set OPTimerButton . Visible to true
  set OPClock . TimerEnabled to false
  set global OPTimer to 30
  set global OPscore to 0

when OPTimerButton .Click
do
  set OPTimerButton . Visible to false
  set OPcanvas . Visible to true
  set OPClock . TimerEnabled to true

when OPTarget .Touched
x y
do
  set global OPscore to get global OPscore + 1
  call OPTarget .MoveTo
  x random integer from 0 to OPcanvas . Width
  y random integer from 0 to OPcanvas . Height

when OPClock .Timer
do
  set global OPTimer to get global OPTimer - 1
  set OPTimerLabel . Text to join "Timer : " get global OPTimer
  if get global OPTimer = 0
  then
    set OPscore . Visible to true
    set OPcanvas . Visible to false
    set OPscore . Text to join "Your Score Is : " get global OPscore
    set OPRReset . Visible to true
    set OPClock . TimerEnabled to false

when OPRReset .Click
do
  set OPscore . Visible to false
  set OPall . Visible to false
  set InfoMainScreen . Visible to true
  set OPRReset . Visible to false

```

OPall

OPcomp

OPTimerButton

OPReset

spacer12

OPTimerLabel

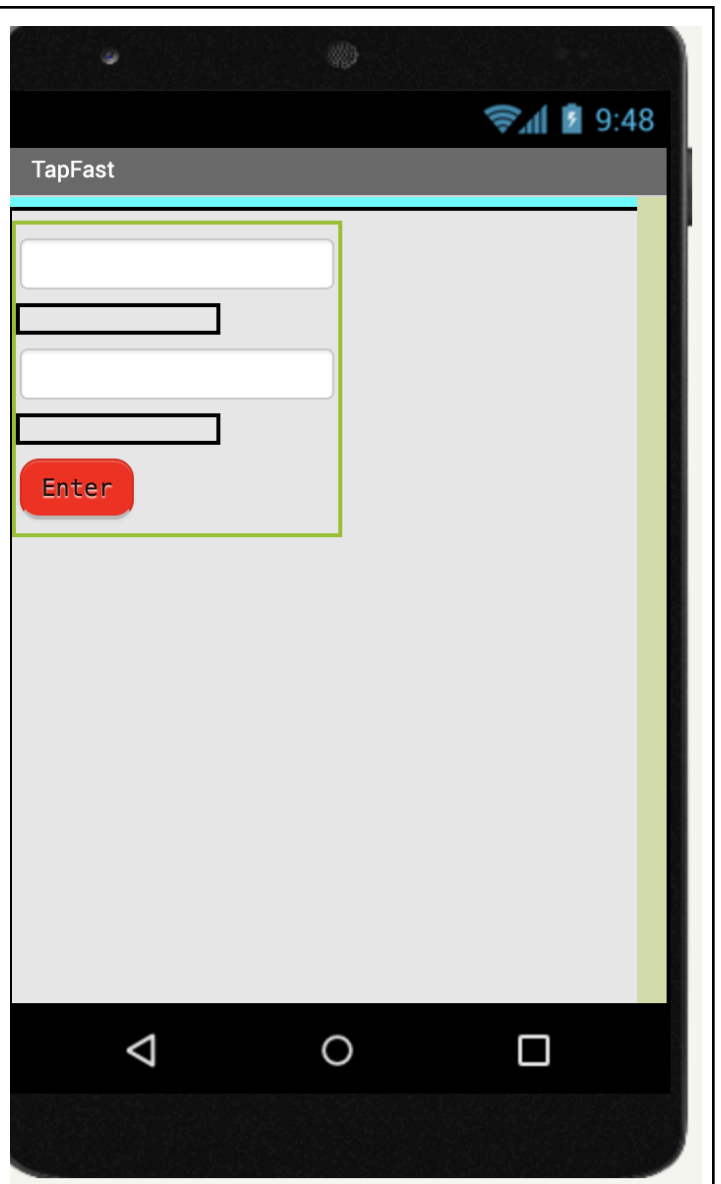
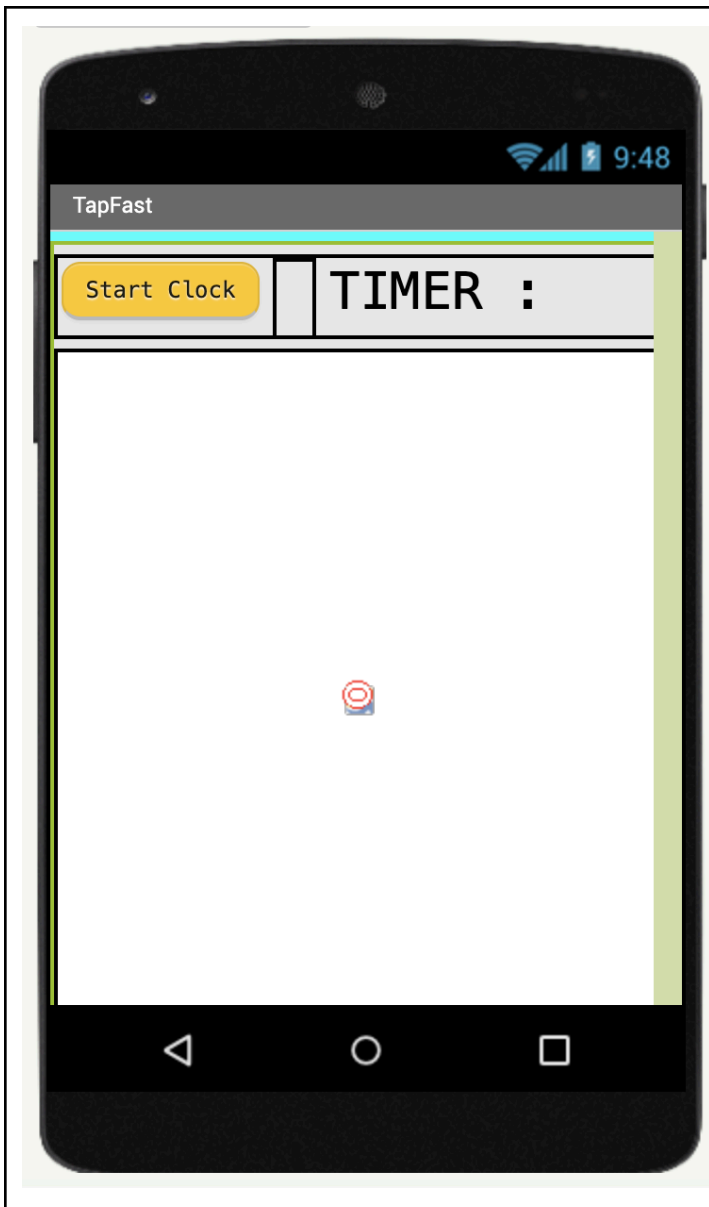
OPscore

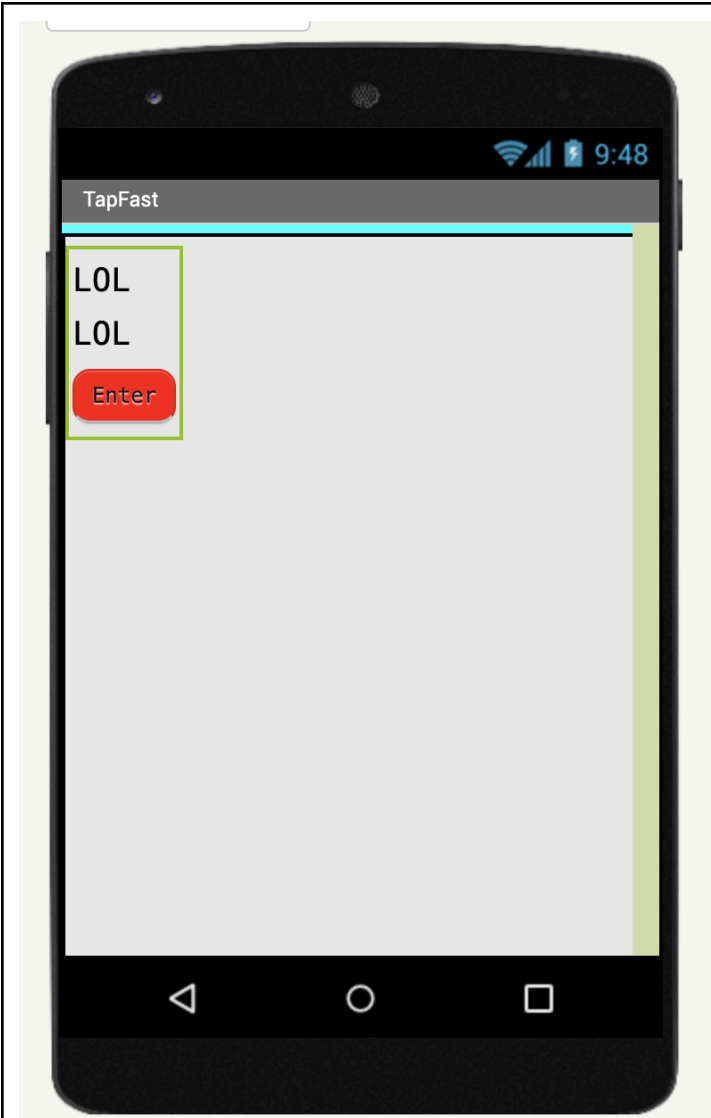
OPcanvas

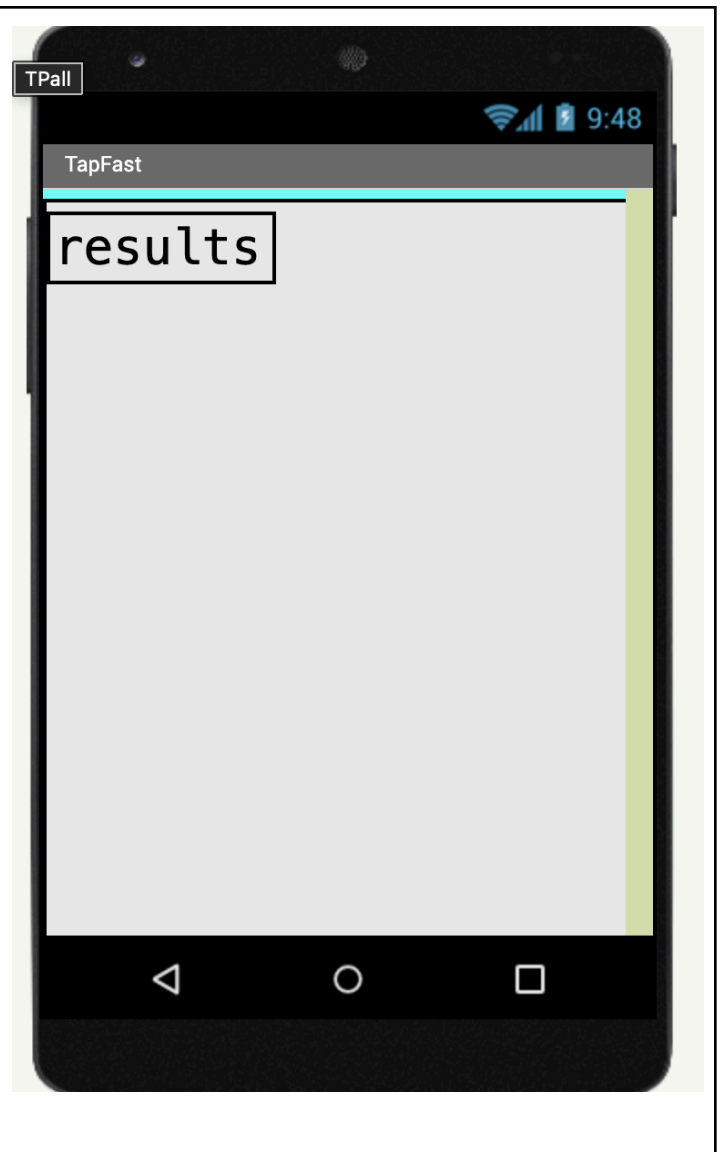
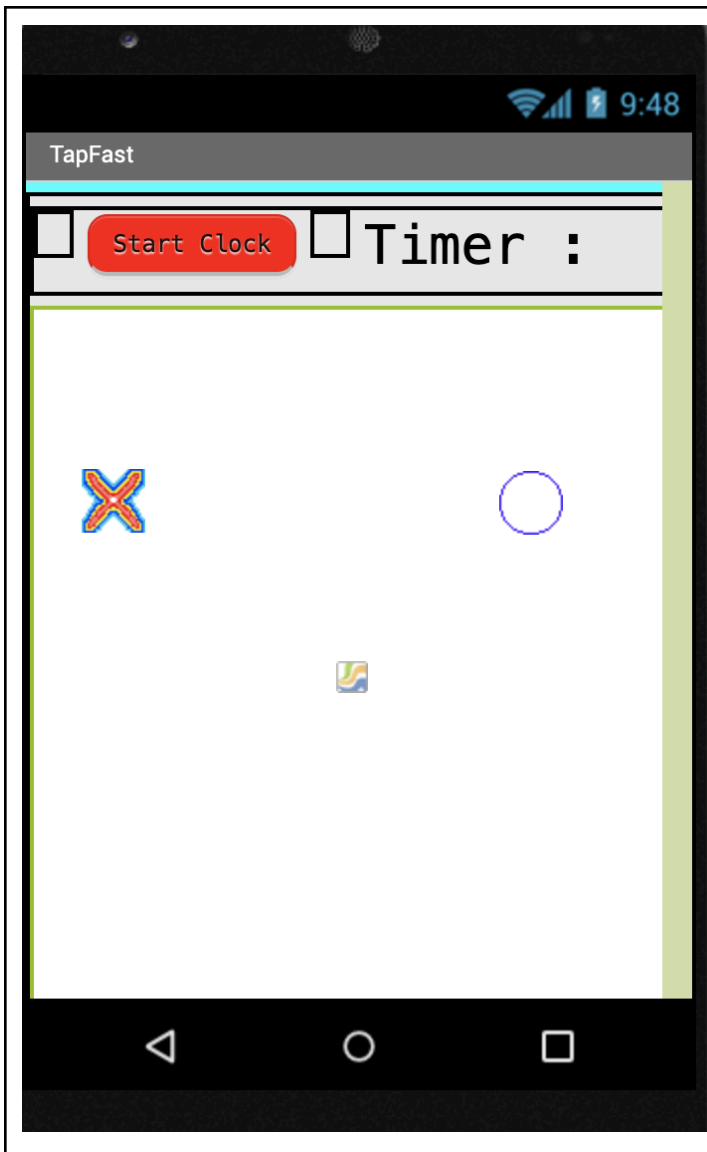
OPTarget




















- OPall
 - OPcomp
 - OPTimerButton
 - OPReset
 - spacer12
 - OPTimerLabel
 - OPscore
 - OPcanvas
 - OPTarget







- [-]  TPall
 -  TPRReset
 - [-]  TPResultsinfo
 -  TPResults
 - [-]  TPcomp
 -  spacer232
 -  TPTimerstart
 -  spacer231
 -  TPTimerLabel
 - [+]  TPCanvas
 - [-]  TPInputs
 -  InputP1
 -  spacer23
 -  InputP2
 -  spacer24
 -  TPInputButton
 - [-]  AssigningXO

```

initialize global TTimer to 30

initialize global xScore to 0

initialize global oScore to 0

when MainTwoPlayer Click
do
  set global xScore to 0
  set global oScore to 0
  set InfoMainScreen Visible to false
  set IPlay Visible to true
  set IInputs Visible to true
  set IPresuits Visible to false
  set IPhaset Visible to false

when IInputButton Click
do
  set IInputs Visible to false
  set AssigningXO Visible to true
  set X text to join InputP1 text
  set Y text to join InputP2 text
  set IPhaset Visible to true

when ITPick Click
do
  set AssigningXO Visible to false
  set ITPick Visible to true
  set ITPick TimerEnabled to false
  set ITPickLabel text to Timer
  set ITPickstart Visible to true

when ITPickstart Click
do
  set global TTimer to 30
  set ITPick TimerEnabled to true
  set ITPickCanvas Visible to true
  set ITPickstart Visible to false

when SP_X Touched
do
  call SP_X MoveTo
  x random integer from 0 to ITPickCanvas Width
  y random integer from 0 to ITPickCanvas Width
  set global xScore to get global xScore + 1

when SP_O Touched
do
  call SP_O MoveTo
  x random integer from 0 to ITPickCanvas Width
  y random integer from 0 to ITPickCanvas Width
  set global oScore to get global oScore + 1

when ITPick Timer
do
  set global TTimer to get global TTimer - 1
  set ITPickLabel text to join Timer
  if get global TTimer = 0
  then
    set ITPick Visible to false
    set ITPick TimerEnabled to false
    set ITPickCanvas Visible to false
    set ITPresuits Visible to true
    if get global xScore > get global oScore
    then
      set ITPresuits text to join InputP1 text
      has
      get global xScore
      points
      White
      InputP2 text
      has
      get global oScore
      points. (If no points shown, consider it as 0 p.)
      InputP1 text
      Wins
    else if get global xScore < get global oScore
    then
      set ITPresuits text to join InputP2 text
      has
      get global oScore
      points
      White
      InputP1 text
      has
      get global xScore
      points. (If no points shown, consider it as 0 p.)
      InputP2 text
      Wins
    else if get global xScore = get global oScore
    then
      set ITPresuits text to join Both of you tied with
      get global oScore
      points
      Visible to true
      set ITPhaset Visible to true
      set ITPhaset Visible to false
      set ITPresuits Visible to false

```

