Analysis of the Brief

Our game has six requirements:

1) It needs to be a 2D game. As the team has decided that we will be using Unity to develop our game, this is a simple requirement to meet.

2) The main control mechanic should be a simple tap. This is a little vague. Should there be one place to tap, multiple or no restriction on where the player can tap? We got clarification that holding a tap

3) The game needs to be multiplayer, with players taking turns. Something to note is that the number of players has not been specified. Related to the last requirement, should the first player tap once then let the other player tap or can they repeatedly tap until the end of their turn?

4) While being multiplayer, the game must take place on a single device. While we could design a game that can be played by many players, some players may be unable to watch the game while it is being played by others if there are too many people around the device.

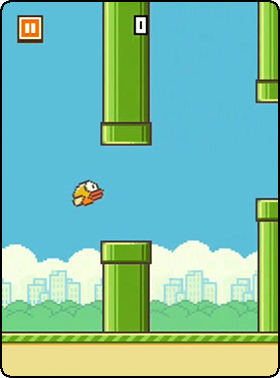
5) The gameplay will be symmetric. This means that all players should be doing the same thing during the game and no player should be given an advantage from the start. If we make a game for two players, for example, and each player controls one side of the screen, the sides should be either identical or (preferably) mirrored.

6) The game needs simple and intuitive rules. As much as we would like to make a complex game, it should be a game that anyone can pick up and play after a brief reading of the controls and aim.

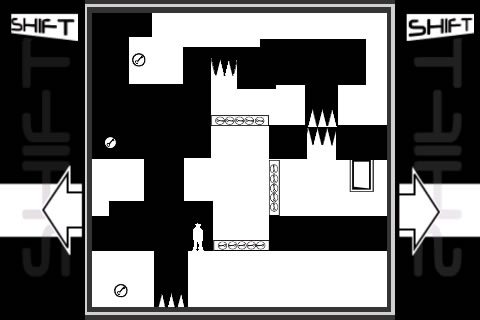
While the "Assignment" slide does not mention whether the game should be competitive or co-operative, the slides following it greatly imply that our client wants us to make a competitive game. Additionally, the next slide implies that the game should be made for a mobile platform.

Single tap games come in many forms:

Clicker games revolve around clicking an object many times to accumulate points, usually with upgrades to make each click worth more points. While a competitive clicker is certainly possible, there would be no turn-based element without one player holding the device for a long time then passing it over for a similar length.

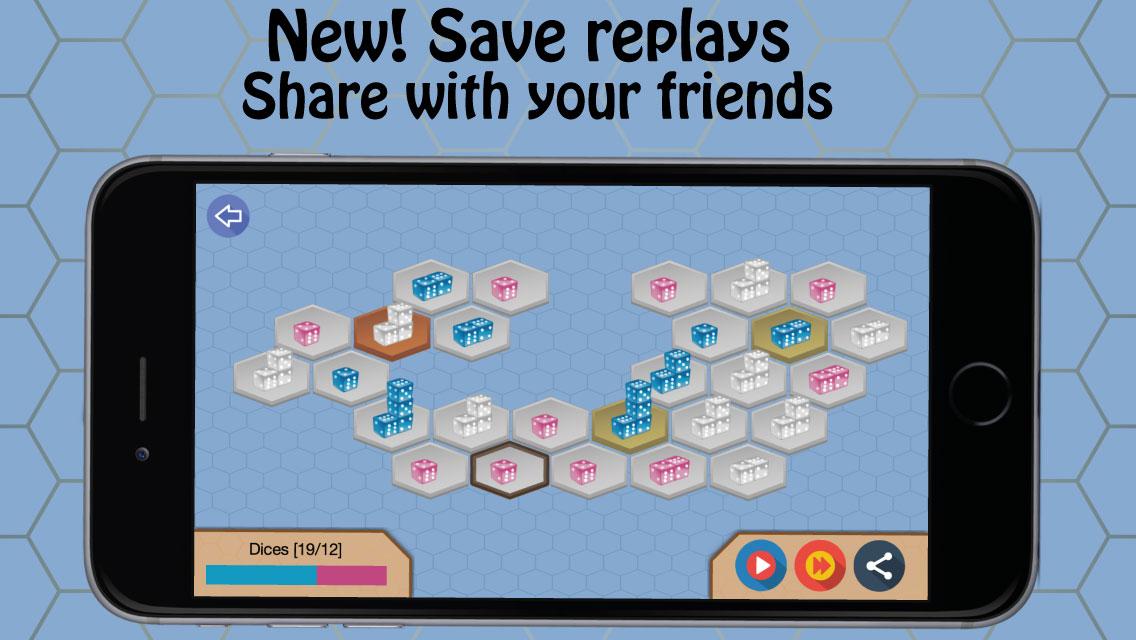
Avoider games involve the player tapping to change their direction in order to avoid obstacles. These can either be endless or consist of a number of courses. For this to fit our brief, one player would have to play followed by the other, not simultaneously. Therefore, an endless avoider would be less appropriate as one player could be playing for a very long time. With a course-based avoider, players would compete to achieve either the fastest time around the course or the most points picked up along the way.

There are many endless runner games that can be turned into one-tap games. In this game, gravity is flipped on a button press. Other games use the single input to control jumping.



This game’s signature mechanic is that whenever the player presses a specific button, the white foreground becomes the background, the black background becomes the foreground and the screen flips upside down. It would be difficult to design levels that work with the one-tap gameplay. This game was ported to mobile devices, so maybe this mechanic could work if the player ran automatically and jumping required a different part of the screen to be tapped. The multiplayer aspect is another issue.

Some puzzle games are designed for single taps. Games like the one pictured right use these taps to remove blocks from the screen. This could be made multiplayer with little difficulty. Players would either play on one, large board or two, smaller boards on either side of the screen. Turn-based gameplay would benefit a game like this since there can be a great deal of strategy in each move.

There are multiplayer strategy games that could be retooled to fit our client’s brief. In the game below, players move stacks of dice around a board in an attempt to claim every space. The value of each die is randomized after a player attempts to take an opponent’s stack, so the player with the largest stack is not guaranteed to win a ‘battle’. This might already meet most of the requirements on the brief.

My preferred solution to the brief is a multiplayer strategy game along the lines of Crush the Castle (pictured at the end). In the version I am proposing, both players have a castle and take turns to shoot at the other player’s using a cannon. The castles will be made of either rectangular or square blocks (depending on what looks better with the physics). Because the brief implies that dragging is not allowed, the first tap will select the angle of the cannon and a second tap will select the power of the shot. This order is important because if the player gets an angle that they did not want, they can compensate for that easier by changing the power. To prevent players from being unable to see the opponent’s castle during their turn, the playing area will be split into three zones: Player 1’s castle, the middle of the map (which may have various objects to either avoid or hit) and Player 2’s castle, going from left to right. Tapping the left or right edges of the screen will move the camera to the next zone in that direction, if there is one.

There will be pickups on the screen, either static (i.e. on a haystack) or moving (i.e. on a cloud, travelling horizontally). These change the type of cannonball being shot out of the cannon. Related to this, there will be different types of block used to build the castles. These will react differently to the different ammunition types.

To determine who the winner is, there will be a health bar for each castle, with an empty bar meaning that a small enough percentage of the castle is remaining / within a given box for the other player to win. An additional win condition will be a horizontal line at a consistent height throughout the game (might be visible to the players at the start of each turn). Once there are no blocks above that line, the player attacking that castle wins.

To shake up the game, we will design many levels with different castles. For symmetrical game play, each player will have the same castle in case there are balancing issues between the castle designs. (though we may create a stage where players can choose one of the preset castles if we have time).

Game loop:

1) Start screen. Tap to move on to 2)

2) Stage select screen. Tap the level you wish to play to move to 3). If the player taps on the custom game, move to 2a)

2a) Custom level select. Player 1 (on the left) picks a castle (with the same castle layout as in 2)). Then Player 2 (on the right) picks a castle and the game moves to 3)

3) Player 1’s turn – Angle select. The camera starts zoomed out to see the whole playing area then zooms into the centre zone and pans left. Player 1’s cannon rotates between 0 and 90 degrees. When tapped, the cannon stops. Move to 3a)

3a) Player 1’s turn – Power select. An arrow or some other indicator appears, pointing out of the barrel of the cannon. When the cannon is tapped, move to 3b)

3b) The projectile is fired in an arc. The camera tracks it when it moves out of a predetermined square, hidden to the player. Once the projectile has stopped moving and all physics stuff is finished e.g. the opponent’s castle has stopped wobbling, move to 4). If either of the win conditions are met, move to 5)

4) Player 2’s turn – This is the same as Player 1’s but flipped horizontally. Player 2 performs the same actions as Player 1 but aiming for Player 1’s castle instead. If either of the win conditions are met, move to 5)

5) The message “Player [] Wins!” is displayed on screen, with the brackets being a placeholder for the winning player’s number. The button below this message will return to the stage select screen. Go to 2)

Other stuff that might be important in future but isn’t an official part of the analysis or proposal:

Unity layers (Furthest back to furthest front):

Background of the whole level

Background of the pick-ups (clouds, haystacks and stuff

Castles, cannonballs



I thought about putting pickups in trees, but I would like the object to disappear when hit, so a tree would look less natural.

Tower defence games are possible under these requirements, but only a specific type. Instead of games like Bloons Tower Defence, where the player buys and places towers, the player would be unable to move and shoot at