Luca Bruno
BA Interactive Media (Year 2)

Flappy Bird

Game Summary:
You control a bird that must fly in between as many pipes of varying heights as possible.

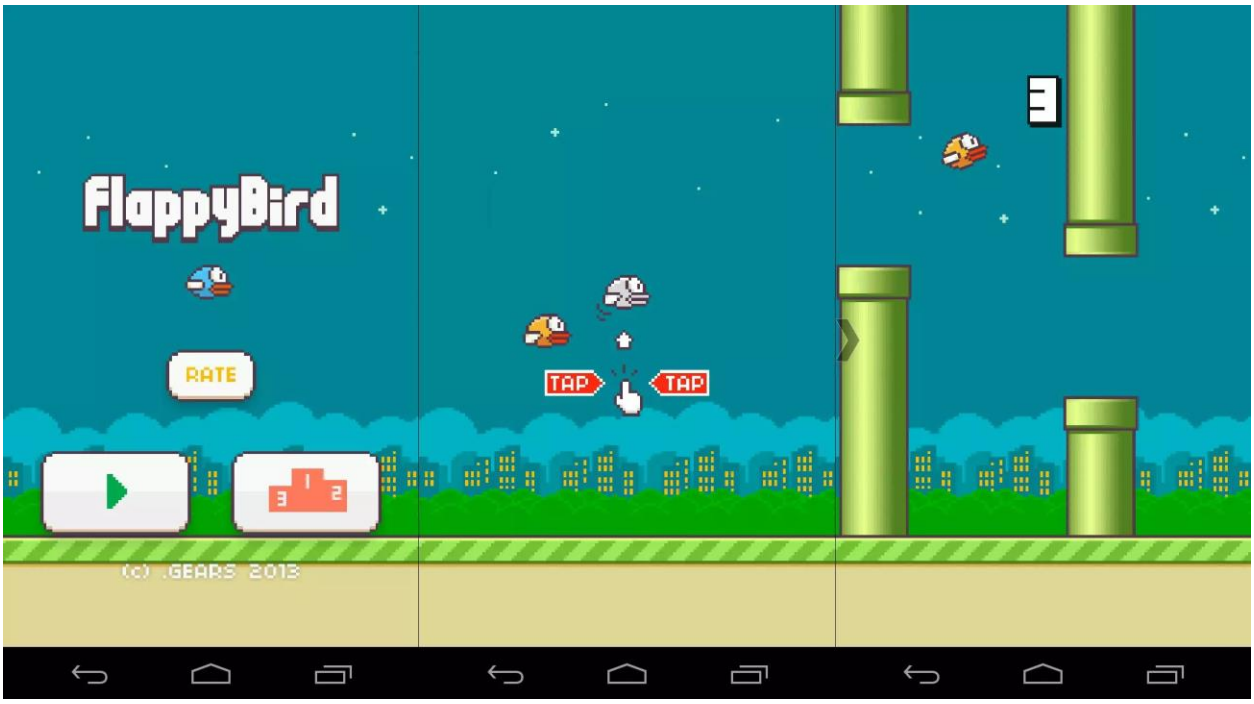
Core Mechanics: List the core features of your game as bullet points.

- The player controls the bird's flying by tapping the screen repeatedly.
- The score is added to each time the bird passes through a set of pipes.
- The player loses when the bird collides with the pipes, or the ground.

Gameplay:
The bird must be kept flying by tapping and traversing through a series of pipes, that will change in height.

Music/audio:
Chiptune styled – classic-sounding, identical to the original. Will feature jumping sound per each tap, coin sound as score increases, and a death sound when the game is over.

Art Style: Will follow the original pixelated 16-bit art style.



About: This is a template that you can add more detail to. A macro design document for a game jam works really well under 3 pages – for this assignment, aim for one or two. Focus on the really important features. The point is to communicate a game idea with your team or think through it yourself so you can dive into prototyping with a plan.

Scope Check

This is the longest and also the most important assignment in the course. It will more or less determine whether you finish with a game you can be proud of. It isn't homework that someone is making you do – it's what happens at the beginning of every successful game project, at any scale. You'll continue to do this for all your games once you've learned how.

The document you create will help keep you on track for the rest of the course so you can **finish your game**.

Write down your answer under each question. There's a lot of text in the questions, but you won't actually need to write much in your answers. You can probably do each section (there are three) in twenty minutes.

Part 1: Visualization and Implementation

Visualization

Start by imagining that you are playing the game. Imagine choosing a specific goal, for example:

- to land on an island in a flying game
- to defeat a turret-type enemy in a space shooter
- to place a tower in the right place to defeat a slow but tough enemy

Answer these questions **based on the moment you have sketched** (not the whole game).

1. Visualize what's on the screen and make a rough sketch of the sequence. Paste a picture of your sketch here:

2. Write down everything you see on the screen. Don't forget things like background art, a score or a timer.

   Bird
   Pipe obstacles
   Ground
   Background art
   Score

3. Write down a list of everything that moves or changes. Are there visual effects, like blinking an enemy red to show damage? Are there sound effects, like a "thump" for placing a tower?

   Bird has flapping animation
   Ground will move with pipes leftwards towards bird
   Bird jumping (sound effect)
   Bird collision/death (sound effect)
   Score increase (sound effect)

4. Write down a separate list breaking down each step of what the player is doing, and what happens in response. How do you tell the game what you want to do? (For example, click to move or press 'E' to charge shrink ray.) How does the game tell you you're making progress (or not)?

   The player is tapping the screen to keep the bird from falling due to gravity. This must be done in the right moments to pass through the incoming pipes. Once a set of pipes have been passed successfully, the game will add 1 point to the score, and a rewarding sound effect will play.

Implementation

1. Imagine how you would write the code for each thing in your last two lists (things that move and player interactions/feedback).
   a. Have you done anything like it before? Is there code from previous projects that you could copy and adapt, rather than starting from scratch?

      This would be my first time using Unity in general, so no.

   b. Do you know of a tutorial or asset pack that could take care of some of the functionality for you? Paste the links here.

      Flappy Bird sprite sheet:
      https://www.spriters-resource.com/mobile/flappybird/sheet/59894/

      Flappy Bird typeface:
      https://www.dafont.com/04b-19.font

   c. Do you have friends or classmates in this course who can help you with these tasks?

      I have multiple friends from my past A-Level course, and even current classmates who attended Level 4 at MCAST Paola that may be able to assist me with getting familiar with the Unity UI.

2. Write down your three biggest coding questions, e.g. “how to make one-way platforms” or “how to make the tower icon move with the mouse”.

   “How to make seamless object loop as an animation”
   “How to add a score counter”
   “How to make restart level button”
   “How to add sound effects for GameObject in prefab”

3. Go to answers.unity3d.com and look for answers for your questions. Did you find answers? Did you generally understand the answers or did they use a lot of unfamiliar vocabulary?

   The URL proved to be very useful when it came to making a restart level button and sound effects. However, questions focused on UI and such were found to be better taught through user tutorials.

4. Make a list of every visual and audio asset in the sequence (i.e. you don't need to think about the whole game). Be thorough – don't forget things like animations and particles, or UI elements like score.
   a. Bird flapping animation 0 sprite
      Bird flapping animation 1 sprite
      Bird flapping animation 2 sprite
      Bird jumping sound
      Bird colliding sound
      Score text
      Score sound
      Pipe Up sprite
      Pipe Down sprite
      Pipes animation
      Ground sprite
      Ground animation
      Background image sprite
      Game title header sprite
      Instructional sprite
      Game over header sprite
      Restart button sprite

   b. If you plan to make any assets, take half an hour and try to make one or two sample assets. Paste the results here. Can you get them to a reasonable degree of quality in that time?

      n/a

   c. If you plan to find some or all of them online, take half an hour and try to find every element in your sequence (environment, character, UI font, etc.) in a matching art style. Try to find a couple of audio assets too. Paste a few of the images here, and note the ones you couldn't find.

Flappy Bird Sound Effects (Best Sound Quality):
https://www.youtube.com/watch?v=w1qyoha3Og8

Flappy Bird Sound Effect:
https://www.youtube.com/watch?v=DMmlXBR5hs4

Part 2: Scale, Challenges and Resources

Scale

Now step back from that sequence mentally and think about the whole game. Think about all the parts of the game that can be numbered and grouped. For example:

- 3 levels
- 5 power-ups
- 2 enemy types
- 6 places to place towers
- 4 foundation piles for cards
- 6 matchable objects for a match-three
- 3 jumping puzzles

1. Make a list. Go ahead and put down a number for each that seems reasonable.

   • No levels (seamless) • No power-ups • 2 “enemies” – Ground, Pipes

2. Mentally cut each number in half. Is the game still playable? Does it still create your core experience? Now try reducing each number to one, and ask yourself the same questions. Write down your final list, with numbers, here.

   List remains the same. The game’s core elements are based on a repetitive and basic game mechanic, which then translates into a simple score system.

Challenges

Based on all of the above, write down the top three challenges you foresee in the process of making your game over the next few weeks. Be specific, and phrase them as questions. These are examples of answers that are too vague to be useful to you:

- “I’m not sure I’m good enough at coding”
- “I don’t know where I’ll get all the art”
- “I might run out of time”

These are examples of useful questions:

- “How can I make my elemental system clear to the player?”
- “How can I tell whether an enemy can see the player?”
- “Can I find pixel-art environment assets to match the characters I’ve found?”

1. Write your questions here.

   “How can I make the pipes height vary randomly?”
   “How can I replicate Flappy Bird’s feel accurately?”

Resources

1. The most important resource is your own time. Look at your calendar for weeks 4-8 of the course (a total of five weeks). For each week, write down the smallest number of hours that you can safely commit to, given your other commitments and interests. Do not assume you can spend every waking hour on your game for six weeks.

   Now subtract 25% of the number for each week because things happen – vet appointments, traffic, hay fever, friends needing favors, accidentally sleeping in, etc.

   Write down the total number of hours for all five weeks here.

Week 1	Week 2	Week 3	Week 4	Week 5
1.5 hours	1.5 hours	1.5 hours	2.25 hours	3 hours
Sprites import Camera setting Bird jump + collide mechanics	Obstacle pipes movement + collide mechanics Looping ground animation	Start menu Game over screen	Score counter Replay button	Sound effects Overall tweaking, refining & testing

Total: 9.75 hours

2. Next, paste in links to three or more specific tutorials that will help you make your game. Don't just write down the top three search results – watch parts of them and make sure they're relevant to what you want to do.

   Make your own Flappy Bird in 10 minutes (Unity Tutorial):
   https://www.youtube.com/watch?v=uRWmEjxY334

   Unity Tutorial How To Make Restart Button To Reload The Scene:
   https://www.youtube.com/watch?v=01sd0VEew7A

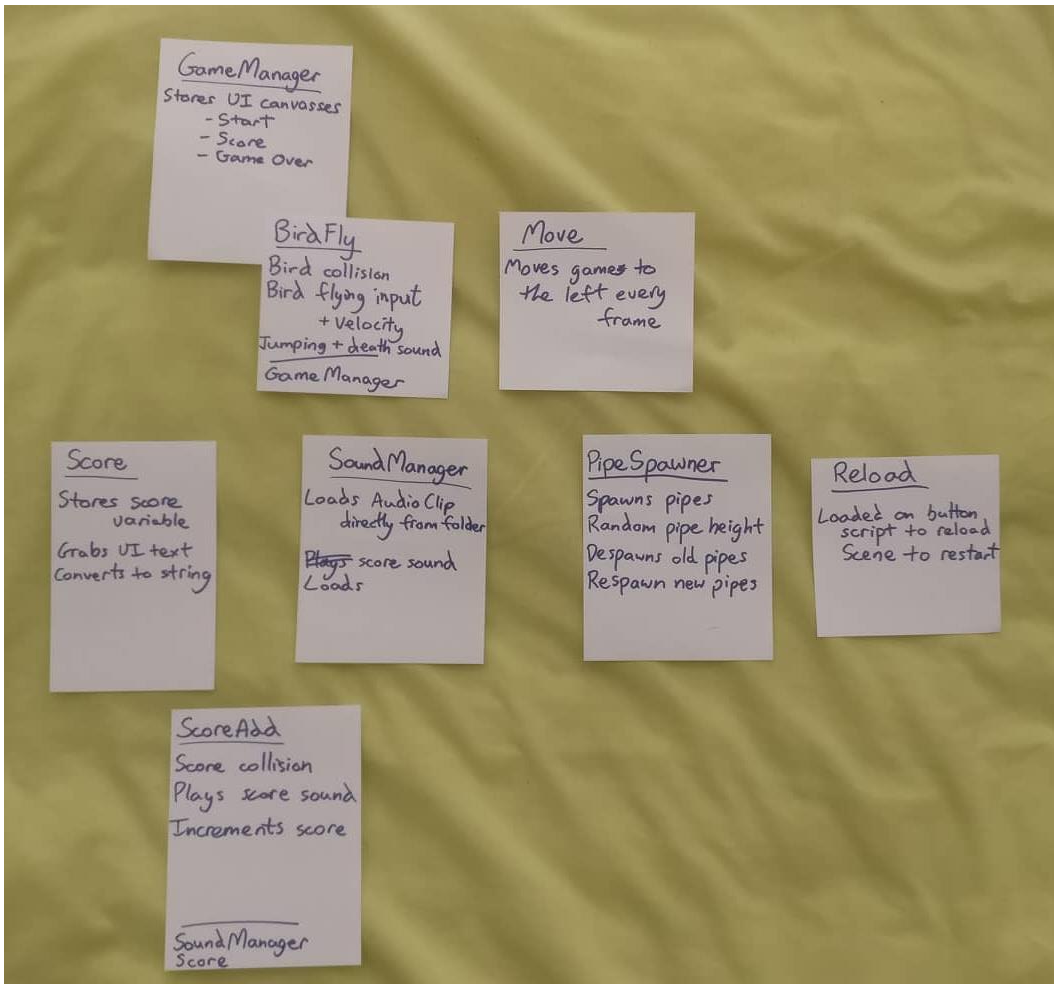
   How to add sound or audio effects SFX to Unity 2D arcade game:
   https://www.youtube.com/watch?v=8pFlnyfRfRc

Part 3: Reality Check

Assessment

1. Look at your challenges and your resources. Do you feel confident you can make a fun game based on this concept by the end of the course? If so, write a sentence explaining why. If not, this is your chance to rethink your choice of game for this course. Go update your concept doc and your answers above – but save your first idea for the future!

I believe I can carry out the above concept regardless of my beginner abilities in Unity, since the coding aspect involved is very minimal and basic. This would also act as a good baseline in order to gauge my skills in view of future coding endeavors.



Doing a scope check is difficult, and not much fun. But it's the foundation of good design. When you know what the core of your experience is, you know what to build first and you won't get lost in building unnecessary features or assets. You'll also end up with a game that feels intentional and elegant, because every element will contribute to the whole.

But most importantly, you're laying the groundwork you will need to **finish your game**.