Luca Bruno
BA Interactive Media (Year 2)

Neurosis

Game Summary:
You control an adventurer that must fight off and survive the many demons that come out in her nightmares.

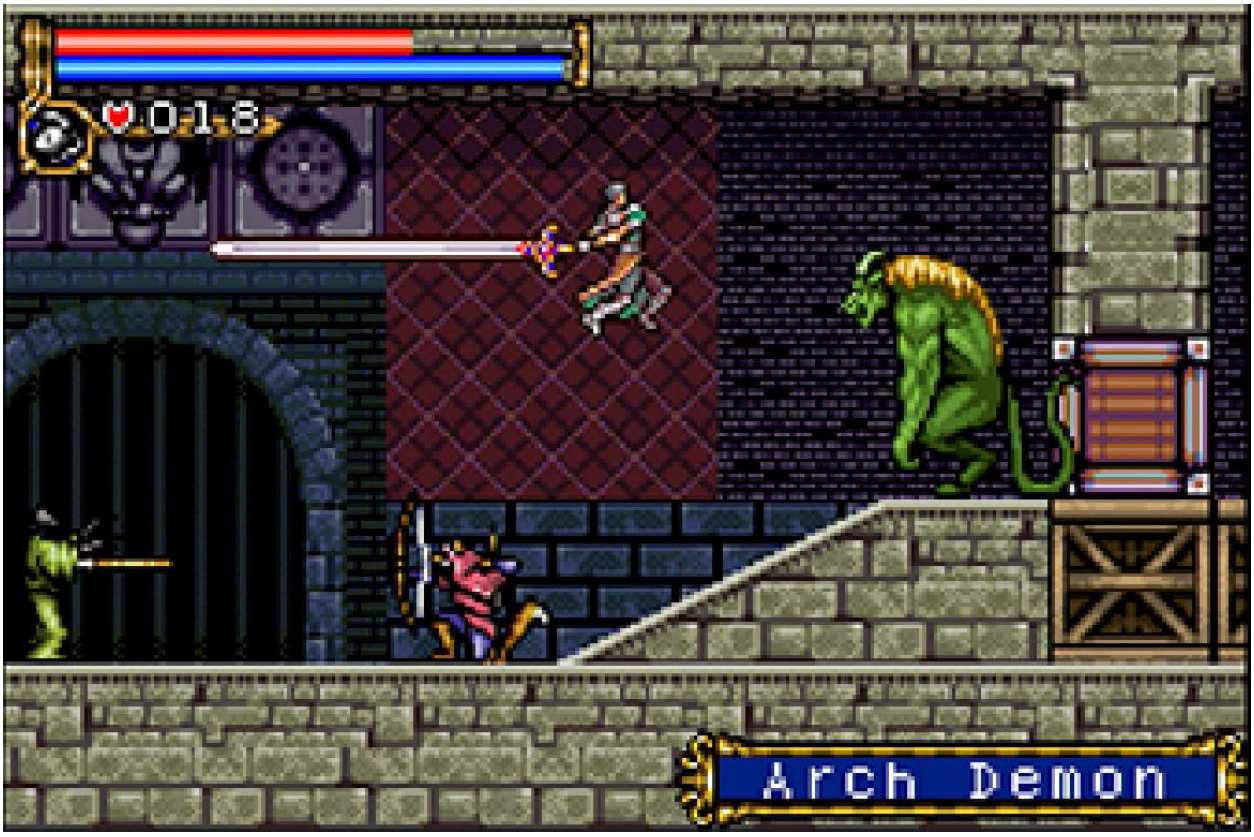
Core Mechanics: List the core features of your game as bullet points.

- The player controls their melee attacks by pressing Q and facing the correct direction.
- The score is added to each time an enemy is killed.
- The player loses when their three lives have diminished.

Gameplay:
The adventurer must be kept alive for as long as possible by jumping, avoiding and attacking waves of enemies.

Music/audio:
Chiptune styled – medieval, haunting, moody and dark. Will feature injured sound per each time the player is injured, and a death sound when the player dies.

Art Style: Pixelated, medieval – similar to games like early Castlevania, for example.



Luca Bruno
BA Interactive Media (Year 2)

Game Engines 1
Phase 2, Task 1

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.

   Protagonist
   Aggressive enemies
   Ground
   Background art
   Health counter

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?

   Protagonist has idle/running/attacking/damaged/death animations (and sound effects)
   Enemies have idle/walking/running/attacking/damaged/death animations (and sound effects)
   Background music playing

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 pressing WASD and Q to traverse the map and attack enemies, respectively. Attacking must be done when close to enemies, before they deal damage. For each slain enemy, the game will add 1 point to the score.

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?

      Considering my previous project was Flappy Bird, and worked a lot differently (even in term of player movement) – I would say it is most likely starting from scratch.

   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.

      Fantasy/medieval backgrounds w/parallax support:
      https://craftpix.net/freebies/free-pixel-art-fantasy-2d-battlegrounds/

      3x Character sprite sheets:
      https://craftpix.net/freebies/free-3-character-sprite-sheets-pixel-art/

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

      As said for my previous assignment, I have friends from my past A-Level course and other Level 6 students from MCAST Paola that may be able to assist me with the workarounds of a 2D platformer using Unity.

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 enemies spawn in waves”
   “How to make enemies hunt for and follow the player”
   “How to make enemies attack the player”

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?

   Unity official forums are very helpful when it come to how to utilize certain functions and methods correctly. However, questions focused on the enemy aggressive AI were found to be better taught through user video 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. Player idle animation sprites
      Player run animation sprites
      Player melee animation sprites
      Player injury animation sprites
      Player death animation sprites
      Player jump animation sprites
      Enemy1 idle animation sprites
      Enemy1 run animation sprites
      Enemy1 melee animation sprites
      Enemy1 death animation sprites
      Enemy2 idle animation sprites
      Enemy2 run animation sprites
      Enemy2 melee animation sprites
      Enemy2 death animation sprites
      Player injury sound
      Player death sound
      Main menu background music
      Game background music
      Background1 sprites
      Background2 sprites
      Background3 sprites
      Heart sprites
      Game title header sprite
      Instructional sprites
      Game over header sprites
      Button background sprites

   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.

Old RuneScape Soundtrack: Cave Background:
https://www.youtube.com/watch?v=zXyTSPgHPcQ

Old RuneScape Soundtrack: Nightfall:
https://www.youtube.com/watch?v=VW6EudU6Lb0

Runescape Man Hit Sound Effect:
https://www.youtube.com/watch?v=YAPCQViWSkw

Runescape Death Sound:
https://www.youtube.com/watch?v=Wu0lVIeL4QI

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 (survival) • No power-ups • 2 varying enemies

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.

   The game would certainly become less varied – however I do believe the game's core element of surviving the longest would still remain present nevertheless.

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 enemies spawn in sporadically?"

"How can I make enemies follow basic AI, but also adhere to animations simultaneously?"

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 hours	2 hours	3 hours	3 hours	2 hours
Sprite research Idea generation on gameplay	Importing sprites Player movement Parallax effect on background	Importing animations Player melee system Enemy AI pathfinding, melee system	Player-Enemy collide mechanics Player health system	UI navigation and menus Sound effects Tweaking & testing

Total: 11 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.

   2D Game Dev Tutorial - Melee Attacking in Unity (Sprites included):
   https://www.youtube.com/watch?v=KamdeKs6eKo

   HOW TO MAKE 2D MELEE COMBAT - EASY UNITY TUTORIAL:
   https://www.youtube.com/watch?v=1QfxdUpVh5I

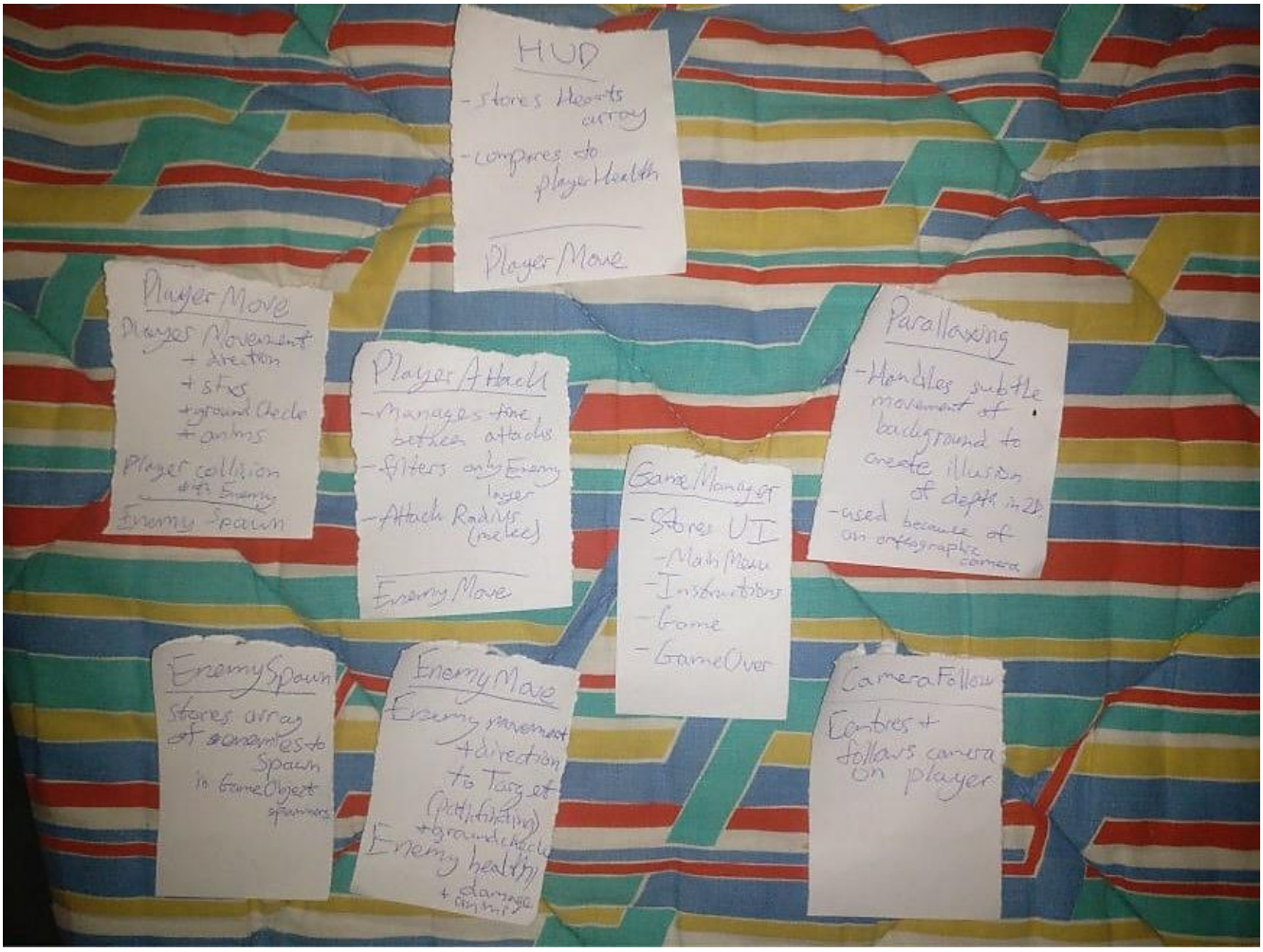
   Game Dev Basics: Let's make a 2D player controller in C# and Unity!:
   https://www.youtube.com/watch?v=44djqUTg2Sg

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 am expecting a much greater challenge from this project, than the last time doing Flappy Bird, since this time around I would be handling a whole lot more animations and interactions all at once; meaning troubleshooting and overall progress would certainly be slowed down dramatically. Nevertheless, I am looking forward to what I may learn.



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**.