FIT3077
Sprint 1 Documentation

Team XDCA

Table of Contents

Team Information

Team Name: XDCA
Team Membership:

Name & ID	Contact Details	Technical & Professional Strengths	Fun Fact
Daniel Gower-Hall, 32504543	Email: dgow0002@student.monash.edu	- Extensive Python Experience - Game Development Related Experience (FIT1048, FIT2099, FIT2102) - Understanding of OO Concepts (FIT2099) - Software Development Process Experience (FIT2101)	Despite my entire family being ethnically British I was born in Hong Kong & lived there for ~5 years.
Xavier Younan 32549814	Email: xyounan@outlook.com	–Game dev experience (pygame) -OOP through FIT2099	I moved to Melbourne from the Gold Coast to pursue a career in game dev 🙂
Liyu chen 31297862	lche0185@student.monash.edu	- Website development - Mobile app development	I been come to melbourne from china for around 6 year by now
Anton Burckhardt 34784926	abur0053@student.monash.edu	- Python flask back-end experience (Klinisches Anwendungsprojekt in Germany) - Understanding of OO Concepts (Datenbanksystem & Informatik II in Germany)	Despite this being my first time in australia for an exchange term, i'm actually ¼ australian

Team Photo:

Team Schedule

Communication Type	**When**	**Information**
Weekly Meeting In Person	5pm Monday	Meetings are conducted with a three phase plan. 1. What have you done 2. Group work 3. Assignment of tasks with due date
Scheduled Meetings On Zoom	Flexible	Same approach to weekly meetings but with an emphasis on group work
Messenger Group Chats	Check at least every 24 hours	General chat, questions, suggestions, planning meetings
Trello	Live Issue Tracking	What are team members currently working on
Google Drive	Document Sharing	Group documents, required links, meeting notes

Workload distribution plan

- Initial tasks will be split up into individual tasks and team members will complete draft copies. Usually a week or two before due dates
- Team will meet to present and discuss completed tasks, group activities will be completed as a team. This meeting is where most of the design decisions will me made
- Polish of individual tasks based on feedback
- Final team meeting to verify project and prepare for submission

Tech Stack

Python
Python was chosen as our programming language as all members of the team had previous experience with it, whereas some had never used Java before. Python is also faster to set up and start testing due as it is a much higher level game engine. This will allow us to prototype and iterate upon our game in a faster manner. Since Python is not a specifically objected oriented only language like Java there may be the possibility for some non OOP code to be written, however this is a worthy tradeoff for the decreased complexity.

Pygame
Pygame is a set of Python modules handpicked for creating video games. It includes tools for audio, rendering, keyboard imports and more. Contrary to popular belief pygame is not a game engine. This will allow the team to handcraft an OOP system, and not have to learn the added complexity that a game engine (such as unity) would add. An alternative library would be Python Arcade, however the popularity of pygame and the increased customization it offers allows more OOP control by our team.

PyInstaller
PyInstall will generate the executable file from the pygame code. Executable generators can be modified quite easily, so it is likely that we may simplify the process (perhaps by using one of the many GUI tools that exist). The current consensus is that a command line tool may give more customisation, and once a working set of commands is created it may be easier to share these among the team (perhaps by adding a generate_exe.bat file to the project).

Optional Modules:
PyOpenGL:
Depending on the level of rendering required PyOpenGL may be utilised to implement the OpenGl API for rendering 2D and 3D vector graphics on the GPU.

PyODE:
Shall physics be required, PyODE provides python bindings for the Open Dynamics Engine (ODE). It is a physics SDK that simulates rigid body dynamics. It supports joints, collision detection, friction and more.

User Stories

Players

As a Player
I want to have my turn clearly indicated to me
So that I know when I should play

As a Player
I want to be able to see all players positions on the board in real-time
So that I can be more engaged and track the games progression

As a first time player
I want an option to view the game rules and objectives at the start of the game
So that I understand the goal and gameplay mechanics required for the game

As a returning Player
I want every new game to have an initially randomised game state
So that every game is a new experience, keeping the game interesting for me

As a Player
I want to be able to flip a chit card of my choosing
So that I may make a strategic choice and progress through the game

As a Player
I want a clearly defined win condition to be stated to me
So that I can understand when the game ends and determine if I have won

Board

As a board
I want to have four clearly distinct dragon tokens
So that the each player can differentiate between their characters

As a board
I want to have clearly marked the progress of each of the dragon tokens on each segment
So that players can follow their progress and it is clear that two players cannot occupy the same tile

As a board
I want to have 4 varying clearly marked animals on each segment of a volcano card
So that players are forced to pick a matching chit card and understand the games response

As a board
I want to have three clearly marked different segments on each of my volcano cards
So that it is clear to the players if a correctly matching chit card was chosen

As a board
I want to have 4 clearly marked starting caves on 4 varying specific volcano cards
So that the initial starting point for each player is clearly depicted

As a newly generated board
I want the order of my volcano cards and starting caves to be randomly generated
So that the game feels fresh and each playthrough offers new strategic challenges to a player

As a board
I want to indicate when a player completes the game by reaching the winning condition
So that the end of the game is clearly communicated to all players

Chit Card

As a chit card
I want to my position to remain constant and clear during the entirety of one game
So that players can strategize and remember where I am in future rounds

As a selected chit card
I want to reveal the value of my card upon a players selection
So that the player understands what selection was made and feels connected to haptics of the game

As a selected chit card
I want to remain revealed throughout one players turn
So that it is clear that the same card cannot be selected twice in one round

As a selected chit card
I want to hide my value if a players turn has ended
So that it is clear to the player that his turn is over and a new players turn may commence

As a chit card
I want my position to be reshuffled after a new game has started
So that each game feels new and each playthrough offers a new challenge to the players

Additional Features

As a chit card
I want to be of special type ‘shuffle’
So that if the player picks me the position of all chit cards is mixed up.

As a chit card
I want to be of special type ‘swap’
So that a player can swap to cards without the other players knowing

As a segment,
I want to be of special type ‘double’ that triggers players moving twice the usual number of segments
So that the game becomes more unpredictable and another layer of complexity is added

As a segment,
I want to be of special type ‘teleport’ that triggers players to teleport to a random location
So that the game become more unpredictable and another layer of complexity is added

As a game master,
I want to customise the number of people playing on setup
So that I can set up the game according to how many people I am playing with

As a game master,
I want to customise the number of volcano cards in the board on setup
So that I can control the expected duration of the game

As a game master,
I want to customise the number of chit cards on the board on setup
So that I can control the difficulty of the game

As a player,
I want to be able to save the game’s progress
So that I can continue playing at a later time

Domain Model

Domain Entity and Relationships

Player: Represents the players in the game.

- Players can flip chit cards
- Players have one specific position

Game Board: Is the table where the game is played

- Stores the layout of the 16 chit cards
- Has 8 volcano cards (4 cave and 4 non cave variants)
- Stores positions of 2-4 players.
- Knows who is the currently active player

Position: Valid location that players can stand on

- One player can be at one position (no duplicates).
- Every position must have one animal type

Segment: Represents a square on the game board where players can move.

- Is a type of position

Cave: Start and end positions of players

- Is a type of position

Animal type: Represents either, Bat, Salamander, Spider, Baby Dragon or Pirate

Chit Cards: Flipable cards in the centre of the volcano.

- They have animal type

Volcano Card: Fixed sections

- Has three segments
- May or may not have a cave

Specific Choices

- We decided not to show the count for chit cards (ie, 1 or 2 or 3 bats). This is because the count would be a member variable for a chit card and thus not relevant for the domain model
- We decided not to differentiate between volcano cards with an indentation for a cave or volcano cards without an indentation for a cave as the domain model would appear simpler and easier to understand, as well as reducing unnecessary complexities further on in the project.
- Pirates are considered an animal type despite not appearing in any position. This could have been modelled differently, but we thought it may introduce unnecessary complexities down the line. It also added flexibilities for multiplicities.

Assumptions

The rules for the game were very clear, so we had no assumptions.

UI Design

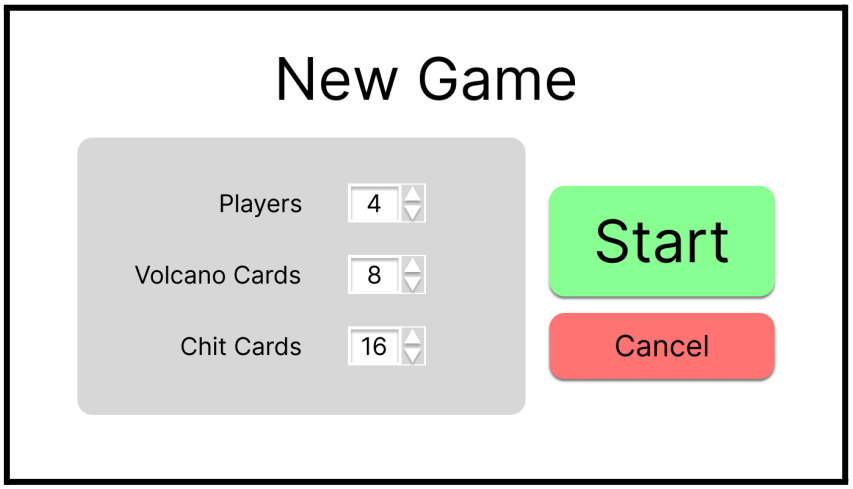
During the Week 3 workshop as a team we created a low-fidelity paper prototype for the to-be-implemented game “Fiery Dragons”. This prototype can be found in the image below:

However, because we will be implementing this game in software this paper prototype does not capture the key differences between a video game and a real board game. Namely, the game setup and method of interaction with a video game will be necessarily different due to the game being interfaced with through a monitor and using a keyboard and mouse in this case.

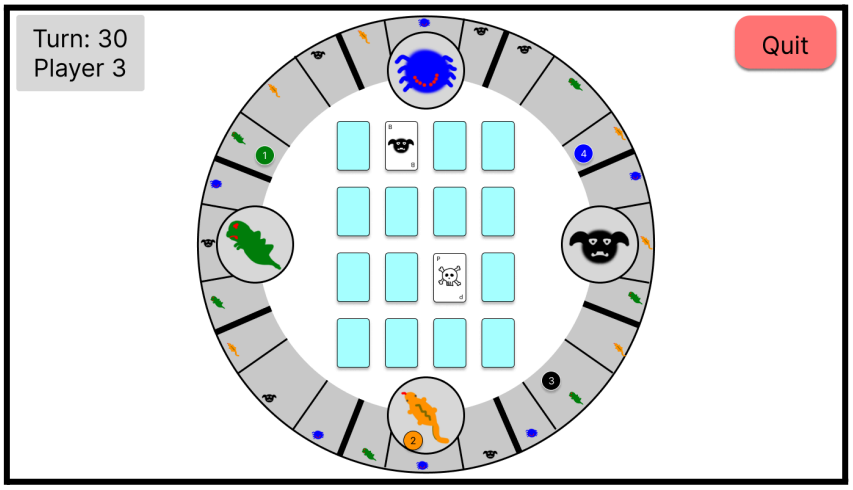
As a result, we have decided to create a second low-fidelity prototype in the popular design tool “Figma”. The design can be found at the following link or seen in the images below:
https://www.figma.com/file/kNWWEORD4ir89f4gk1SnYZ/Game-Design-Sprint-1

Main Menu:

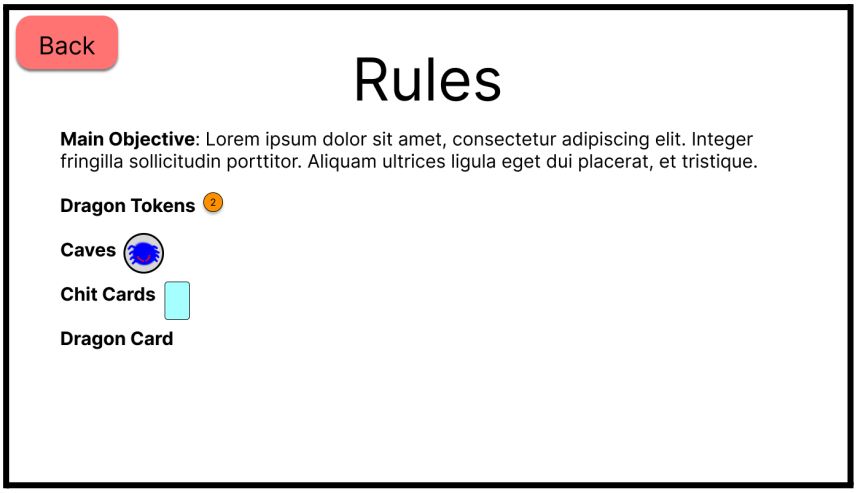
Game Setup:



Game Screen:



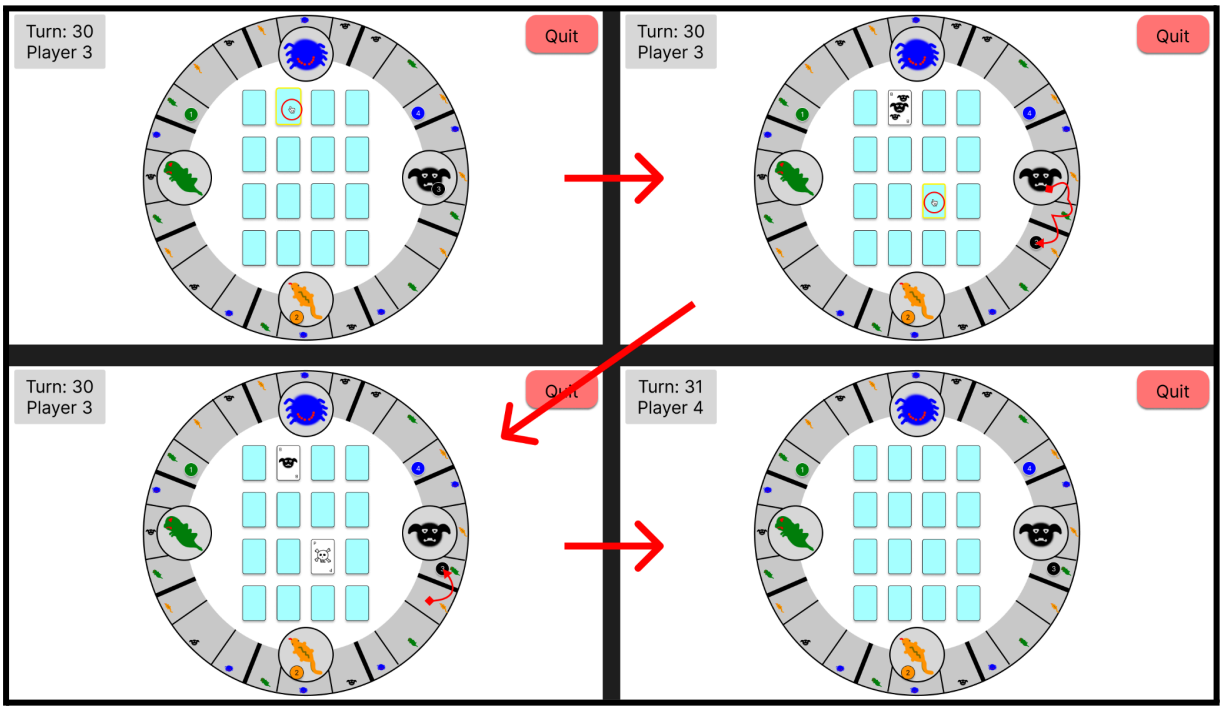
Help Rule Screen:



Notably, being a low-fidelity prototype still, this design does not include stylistic choices, such as fonts, colour scheme, and sprites, which will be made later in the design process after we have assessed what is possible in our chosen tech stack (during Sprint 2).
What has been considered is several key design principles to ensure good usability. These include Donald Norman's Principles of Interaction Design (Norman, 2013), Ben Shneiderman's Eight Golden Rules of Interface Design (Shneiderman et al., 2018), and Gestalt Principles (The Gestalt Principles, n.d.).

For example, with regards to Donald Norman's Principles we have maintained consistency within and across frames/scenes by using standardised shapes and colours for buttons. For Ben Shneiderman's rules we keep users in control by making every interaction user driven; in our case most interactions are click-driven. And in terms of Gestalt Principles, concepts such as a "Common Region" and "Proximity" are displayed in the game setup form to provide clarity on related options, fields, labels, and inputs.

Also included in the Figma design file is an outline of the main gameplay loop through a series of depicted user interactions showing the completion of a full turn at some point during the game. This can also be seen in the image below (note that the red highlights are not intended to be included in design and are shown for clarity):



Other interactions such as game creation are intended to be implicitly explained by the static design. The Figma design file also contains a number of notes on various elements giving justification of inclusion or indications on how to develop into a higher fidelity design for our use going forward.

References

Norman, D. (2013). *The design of everyday things*. New York : Basic Books.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2018). *Designing the User Interface* (6th ed.). Harlow: Pearson Education, Limited.

The Gestalt Principles. (n.d.). Retrieved March 22, 2024, from https://www.interaction-design.org/literature/topics/gestalt-principles