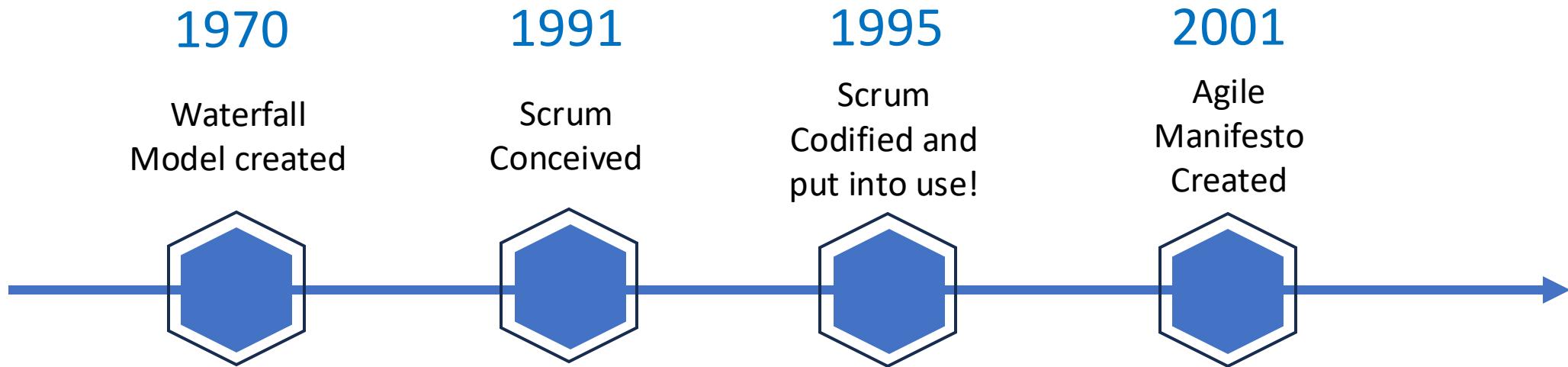


Software Engineering

From Waterfall to Agile



Waterfall to Agile



SOFTWARE ENGINEERING

**Report on a conference sponsored by the
NATO SCIENCE COMMITTEE**

Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer

Co-chairmen: Professor L. Bollett, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

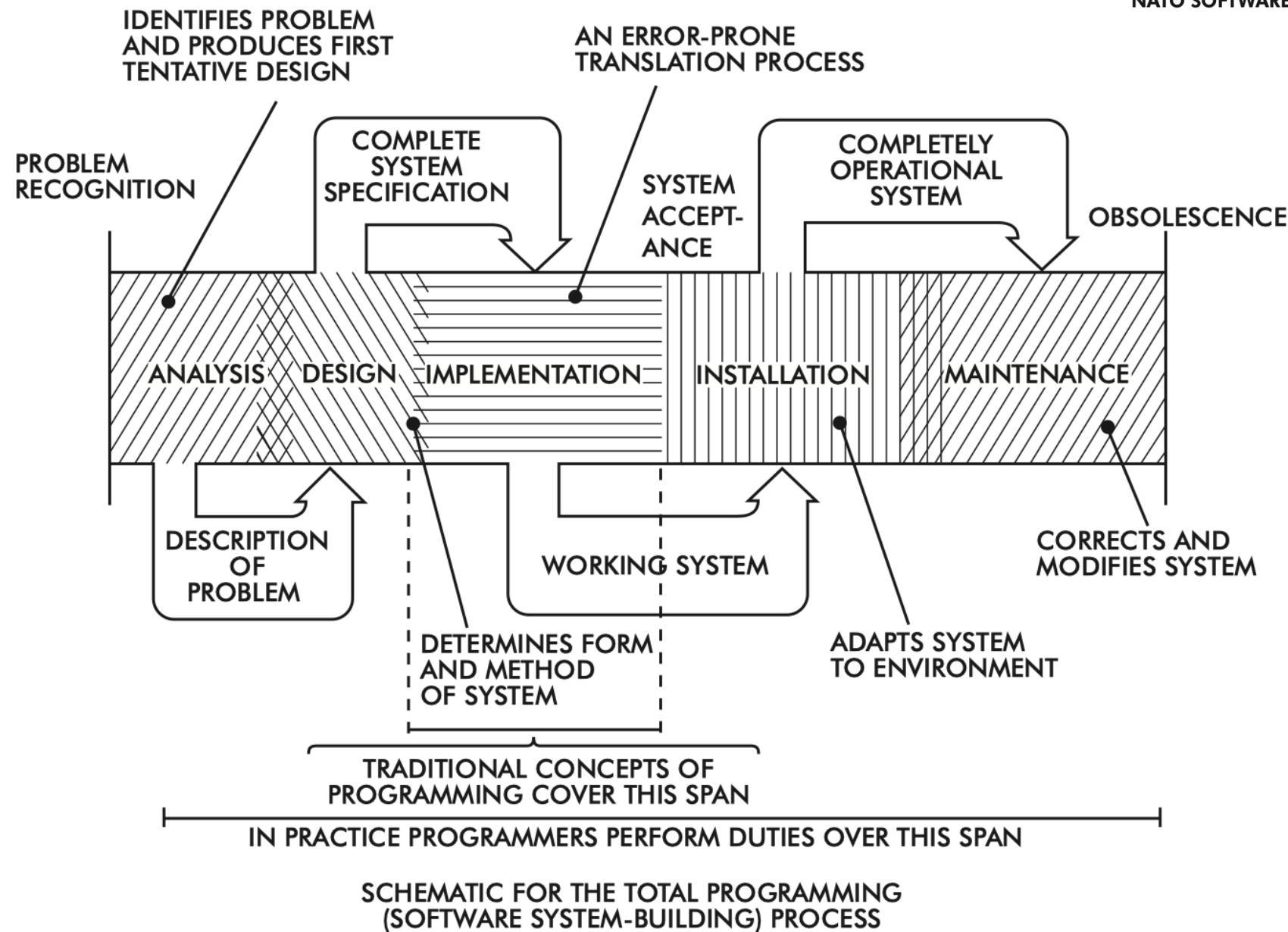
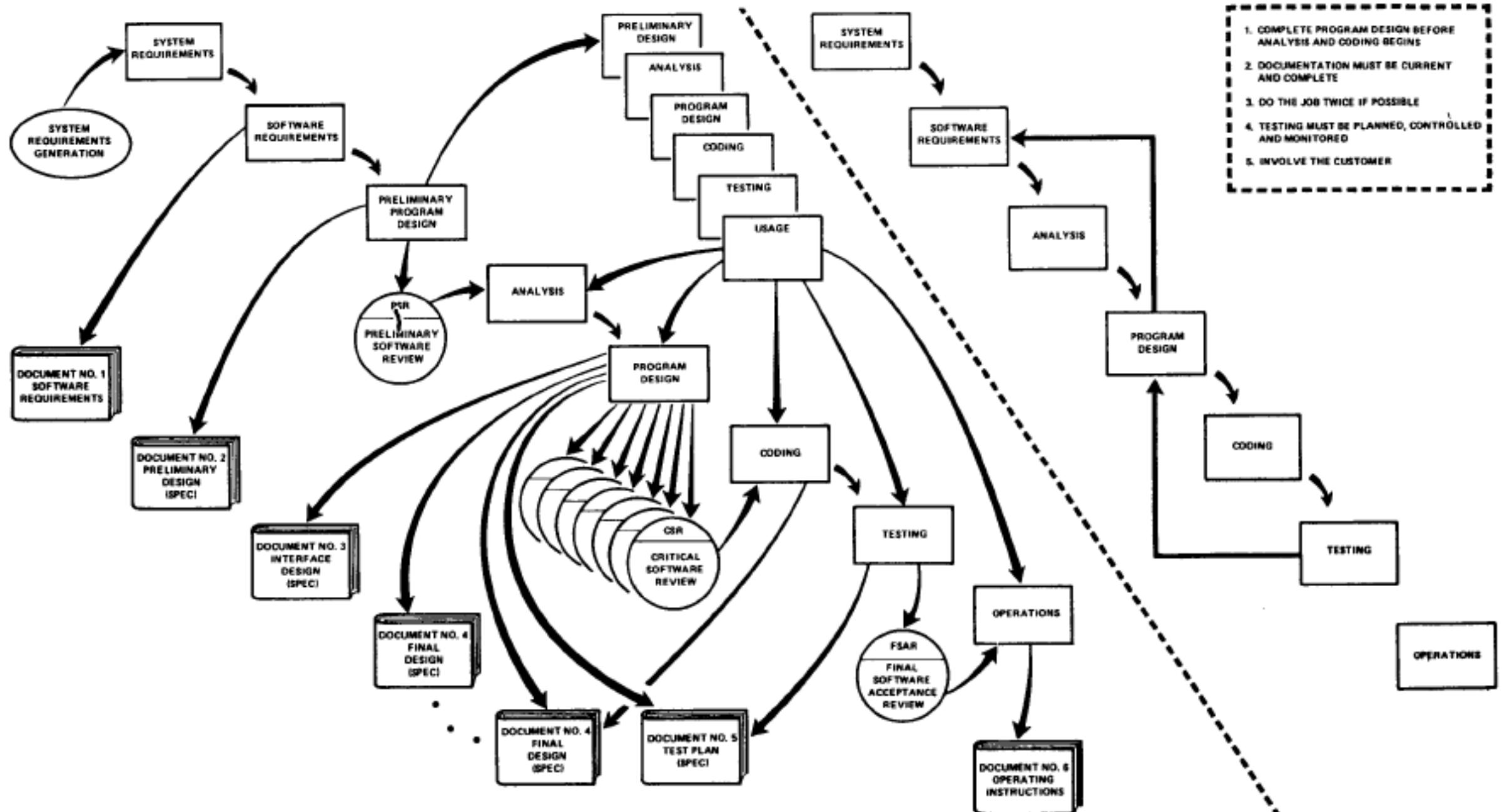
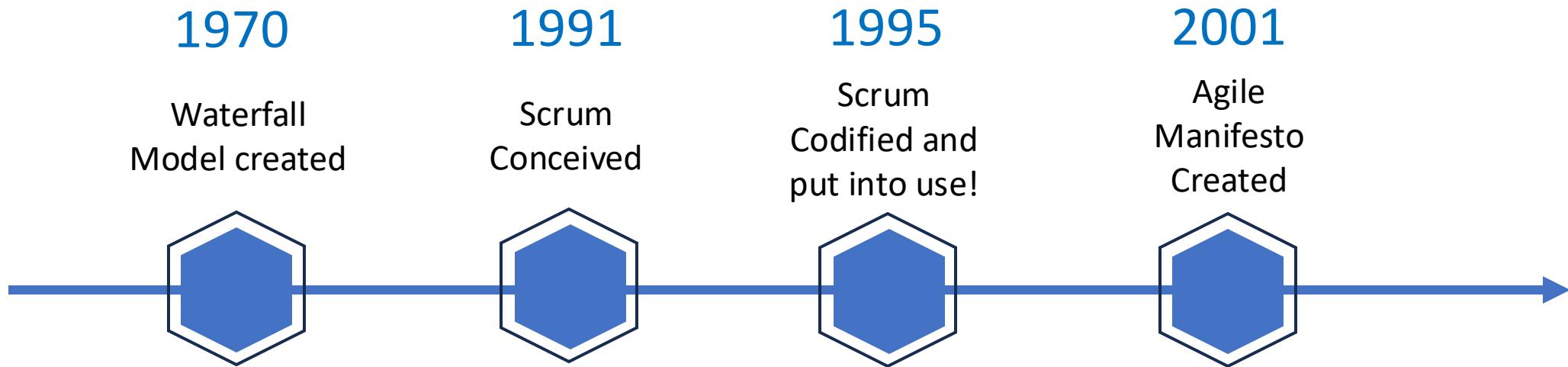


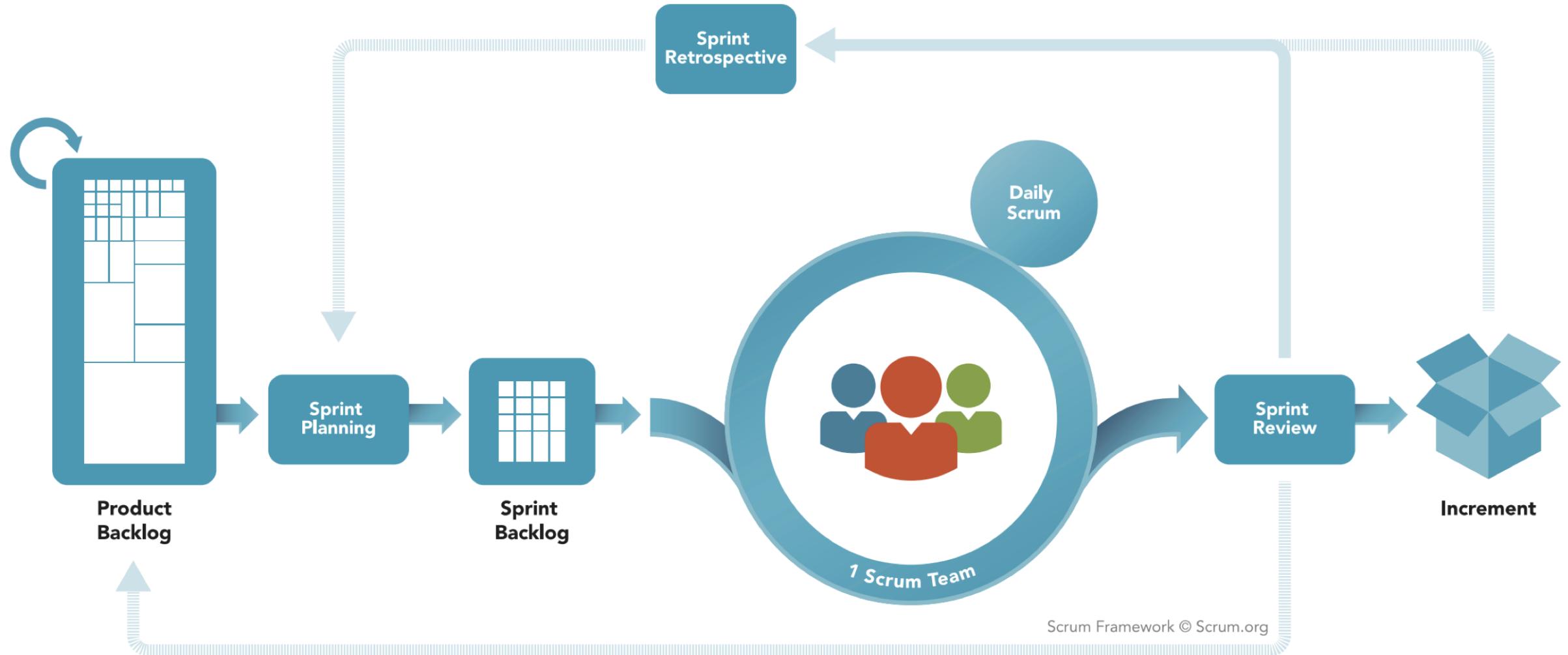
Figure 2. From Selig: Documentation for service and users. Originally due to Constantine.



1970 Winston W. Royce's Final model, published in 'Managing the Development of Large Software Systems'

Waterfall to Agile







COURAGE

Scrum Team members have courage to do the right thing and work on tough problems

FOCUS

Everyone focuses on the work of the Sprint and the goals of the Scrum Team

COMMITMENT

People personally commit to achieving the goals of the Scrum Team

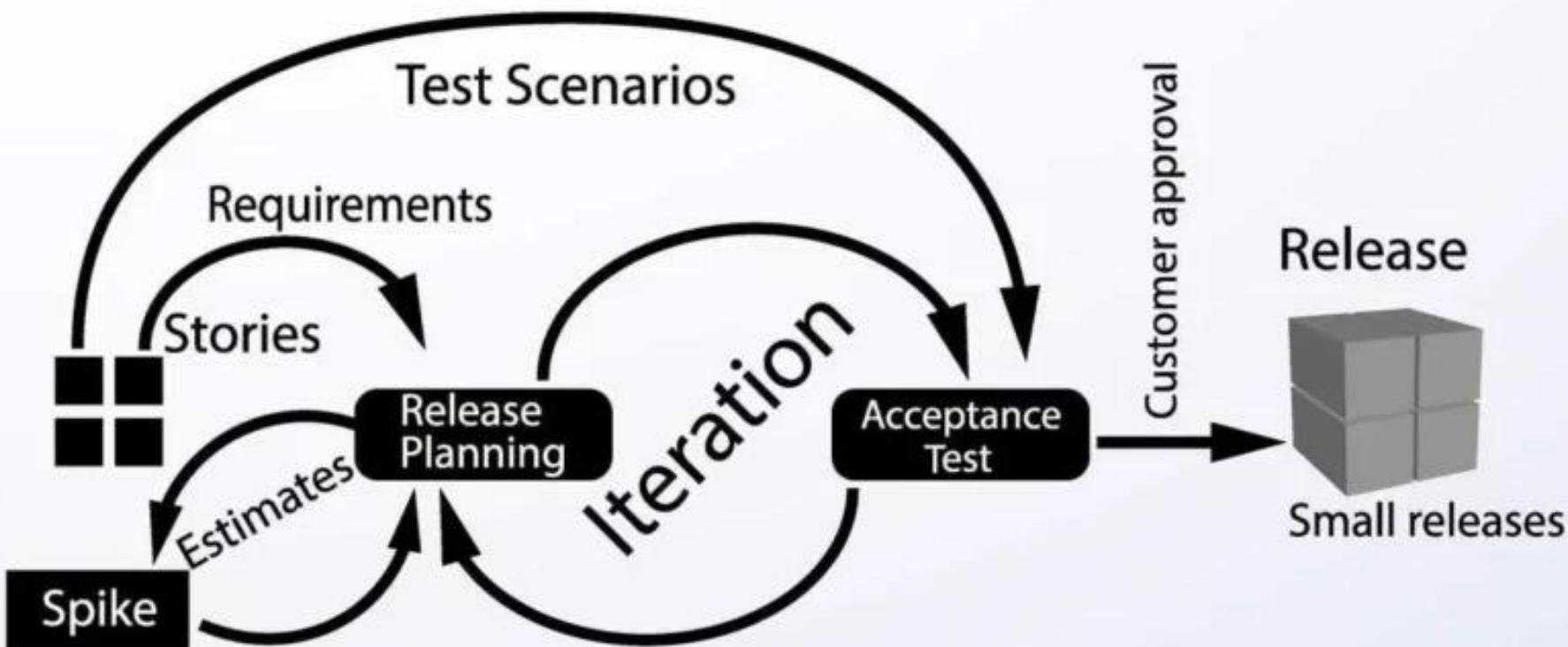
RESPECT

Scrum Team members respect each other to be capable, independent people

OPENNESS

The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work

Extreme Programming



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck

Mike Beedle

Arie van Bennekum

Alistair Cockburn

Ward Cunningham

Martin Fowler

James Grenning

Jim Highsmith

Andrew Hunt

Ron Jeffries

Jon Kern

Brian Marick

Robert C. Martin

Steve Mellor

Ken Schwaber

Jeff Sutherland

Dave Thomas

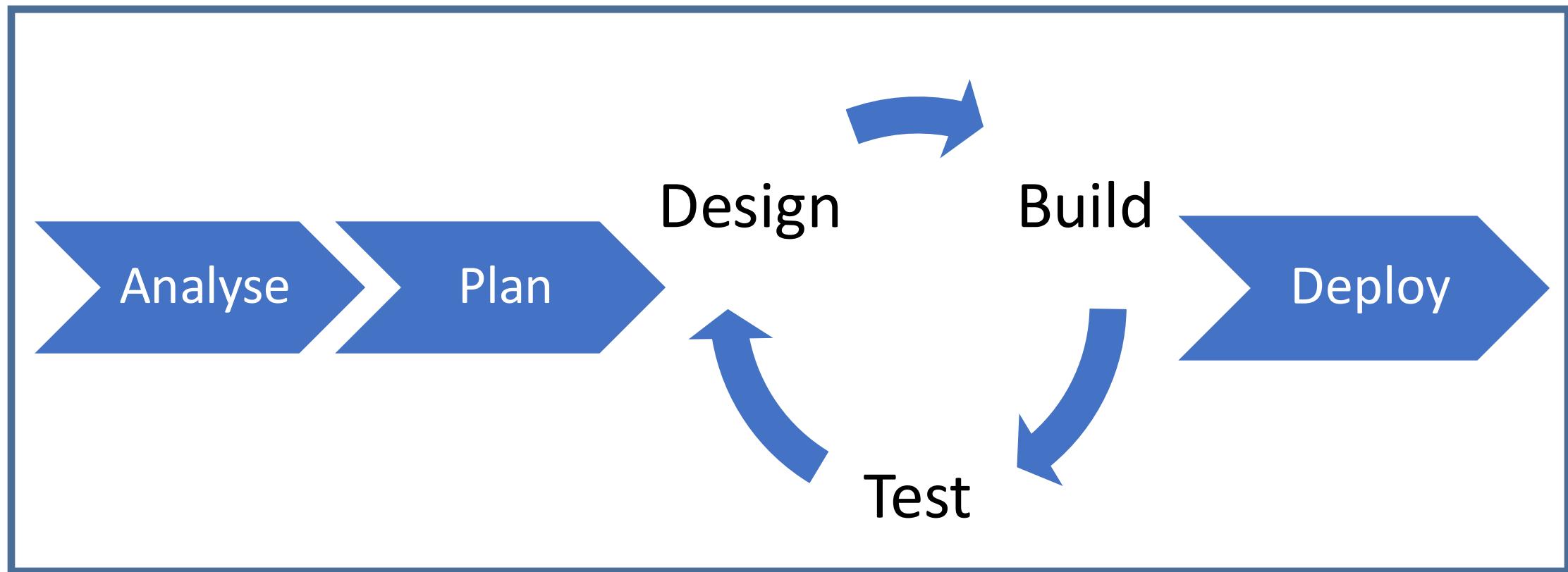
Principles behind the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Principles behind the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

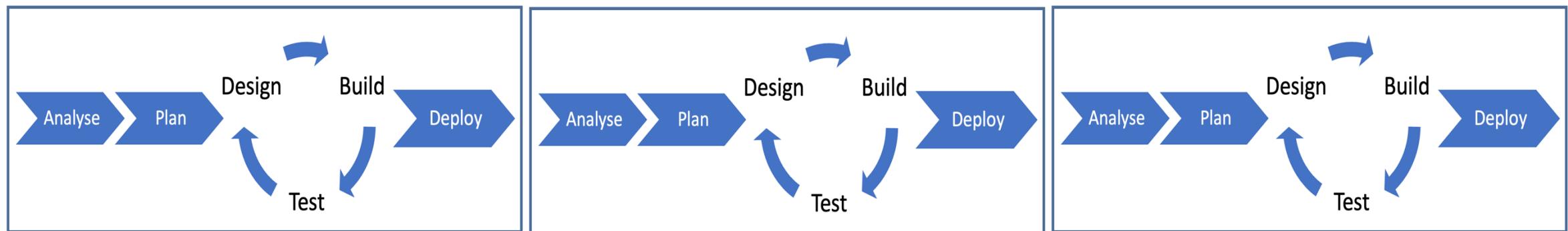
Implementation: a typical ‘sprint’



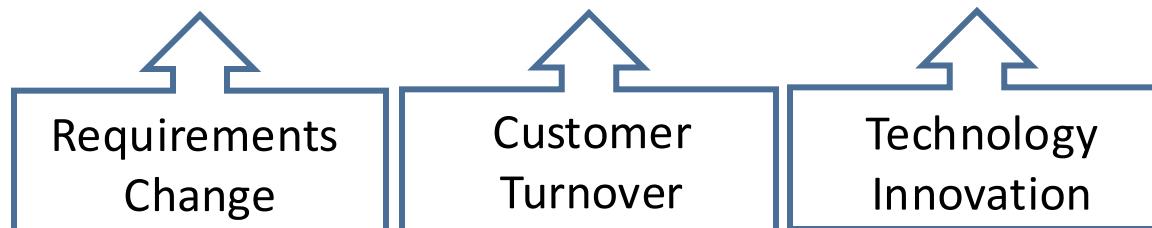
Waterfall



Agile



Project Timeline



Example Game Requirements (Space Invaders)

- The Defender should move left and right and be able to fire bullets
- Invaders must alternate direction as they move down the screen
- Invaders speed of travel should increase as fewer remain and levels progress
- Shields should crumble as they shot by both the Invader and the Defender

Space Invaders

Nicholas Day edited this page now · 1 revision

Requirements

- The Defender should move left and right and be able to fire bullets
- Invaders must alternate direction as they move down the screen
- Invaders speed of travel should increase as fewer remain and levels progress
- Shields should crumble as they shot by both the Invader and the Defender

A large, semi-transparent watermark of the GitHub logo is centered on the slide. It features a dark purple octocat icon with a white outline, set against a light gray background.

GitHub

A man with dark hair, glasses, and a mustache is sitting at a desk, looking down at his hands which are clasped together. He is wearing a blue jacket over a red and white striped shirt. The background is a blurred office environment.

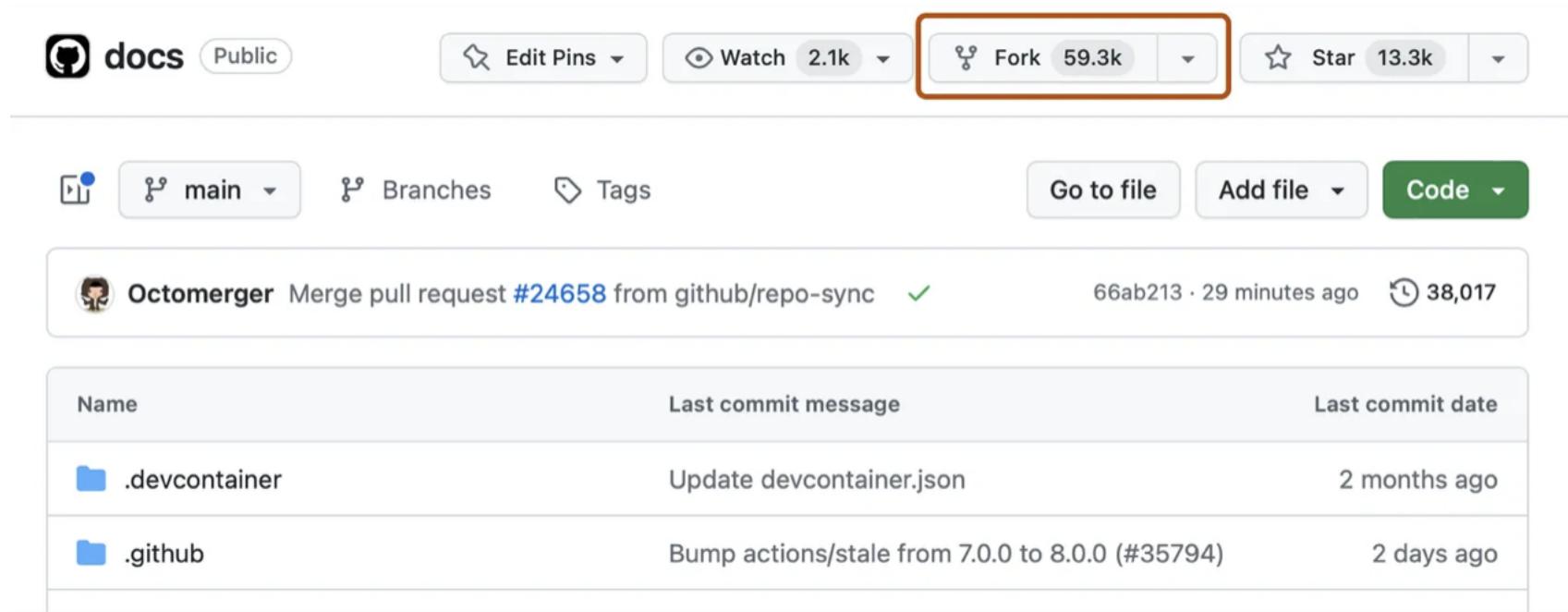
What is
GitHub?

Forking or
branching?



Forking

- Creates a local copy of the entire repo (by another author)
- Maintains the commit history from the original author



The screenshot shows a GitHub repository page for 'docs'. At the top, there are buttons for 'Edit Pins', 'Watch' (2.1k), 'Fork' (59.3k), and 'Star' (13.3k). The 'Fork' button is highlighted with a red box. Below the header, there are navigation links for 'main' (selected), 'Branches', and 'Tags'. On the right, there are buttons for 'Go to file', 'Add file', and 'Code'. A recent commit by 'Octomerger' is displayed, showing a merge pull request from 'github/repo-sync'. The commit message is 'Merge pull request #24658 from github/repo-sync' with a green checkmark, and it was made at '66ab213 · 29 minutes ago' with '38,017' views. Below the commit, there is a table showing two files: '.devcontainer' and '.github'. The '.devcontainer' file was last updated '2 months ago' with the message 'Update devcontainer.json'. The '.github' file was last updated '2 days ago' with the message 'Bump actions/stale from 7.0.0 to 8.0.0 (#35794)'.

Name	Last commit message	Last commit date
.devcontainer	Update devcontainer.json	2 months ago
.github	Bump actions/stale from 7.0.0 to 8.0.0 (#35794)	2 days ago

<https://docs.github.com/en/get-started/quickstart/fork-a-repo>

Branching

- Portion of the repo to isolate work on an issue
- Have to ‘merge’ branches later

The screenshot shows a GitHub repository settings page for the 'docs' repository. At the top, there are tabs for 'Code' and 'Issues' (which is highlighted with a red box), 'Pull requests' (47), and 'Branches' (21). Below the tabs, there are buttons for 'main' (with a dropdown arrow), '21 branches', and '1 tag'. On the right side, there are sections for 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development' (with a red box around the 'Create a branch' button). A note below the 'Development' section says 'for this issue or link a pull request.' The top right corner shows 'Star 13.3k' and 'Settings'.

<https://docs.github.com/en/issues/tracking-your-work-with-issues/creating-a-branch-for-an-issue>

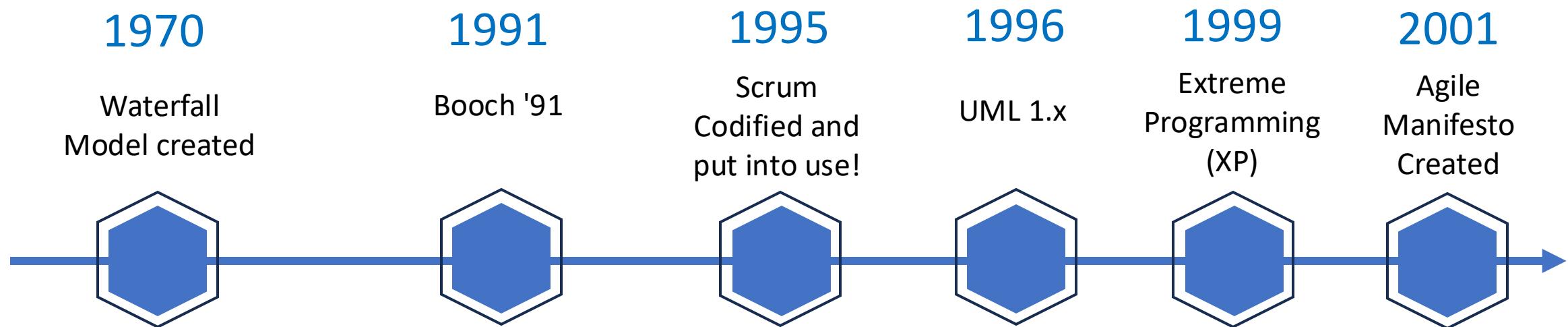


Design

**UNIFIED
MODELING
LANGUAGE**™



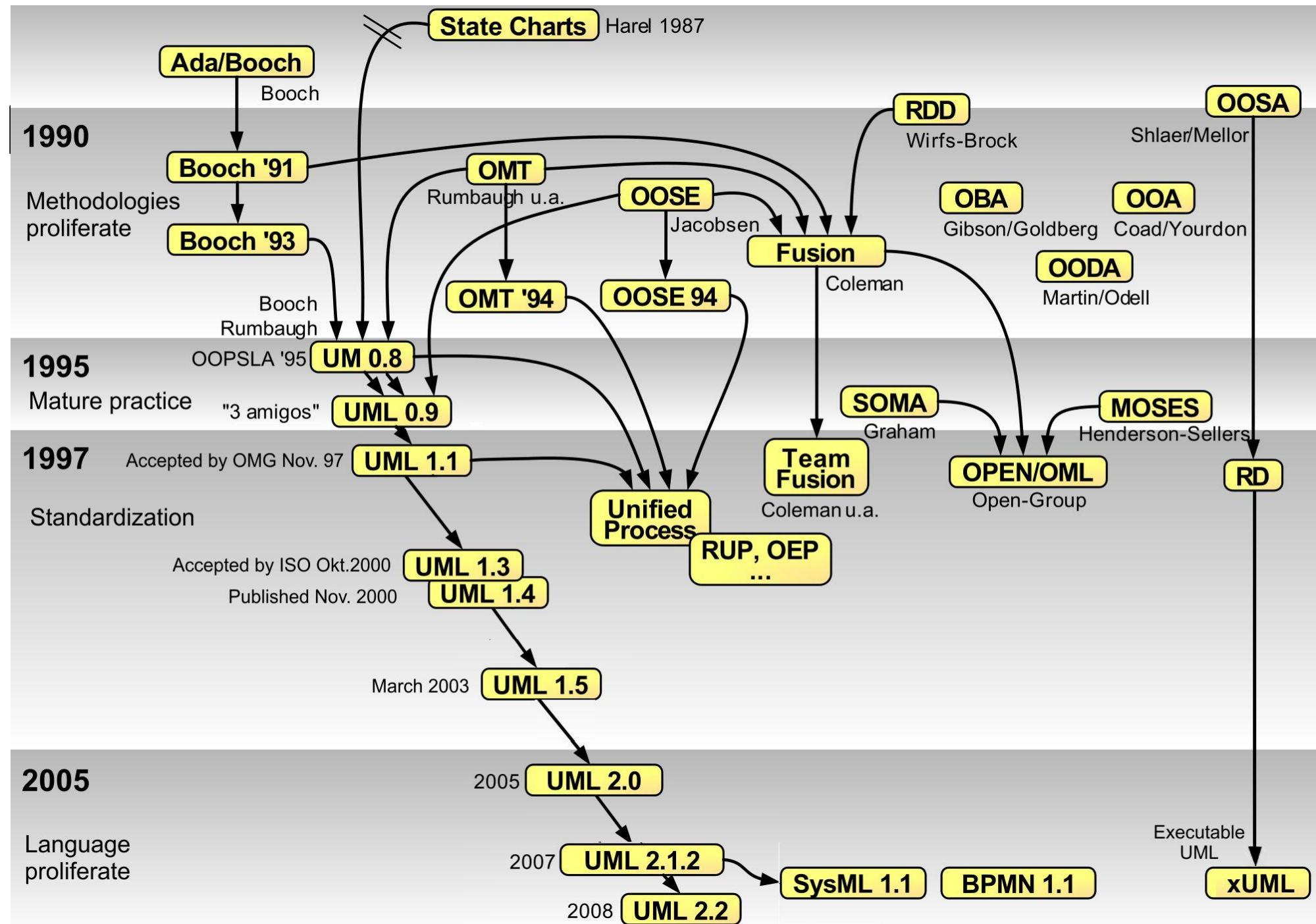
Waterfall to Agile



Unified Modelling Language

- UML developed throughout the 1990s, with UML 1.x published in 1996
- Introduced to standardise the variation in quality of designs in the industry.
- UML provides standard notation, roughly divided into three groups:
 - Behavioural diagrams (e.g. Use Case Diagram)
 - Interaction diagrams (e.g. Sequence Diagram)
 - Structure diagrams (e.g. Class Diagram)
- 1997: UML adopted as a standard by the Object Management Group (OMG)

U



Use Case Diagrams



Use Case Diagram

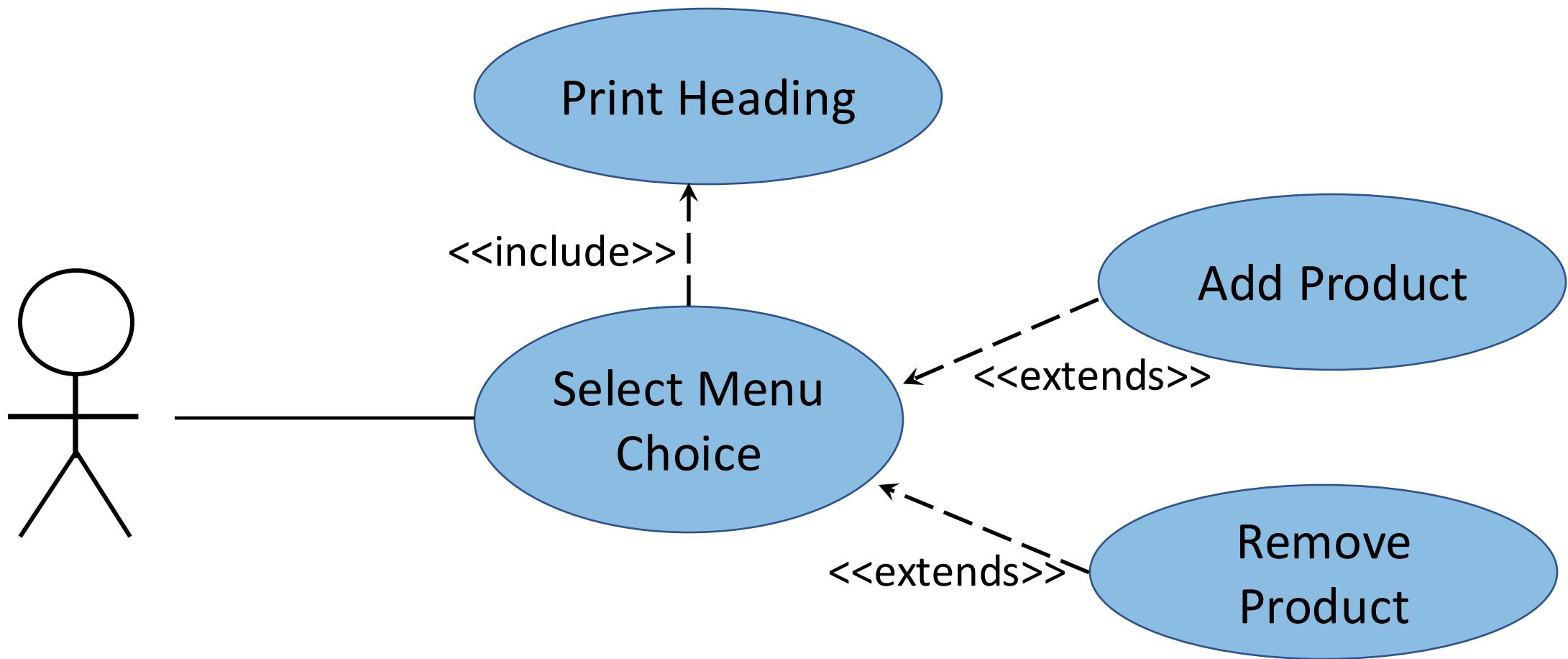
Use Case Diagrams can be used to represent users of a system (actors), and what behaviours they have (methods) in relation to the system (the ‘use cases’).

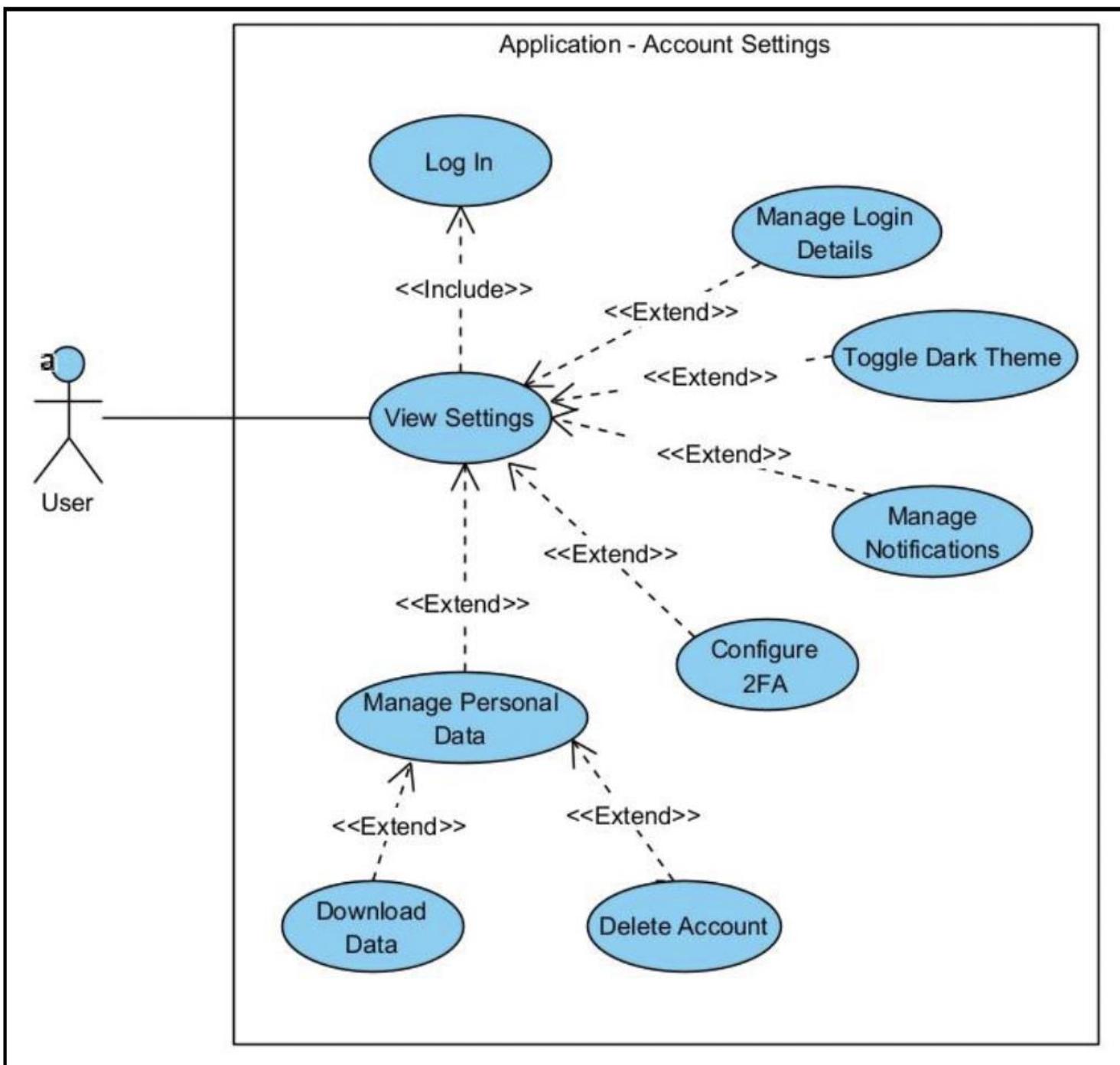
Use Cases (ovals) start with a verb (as methods should do).

<<include>> is a mandatory Use Case (will always happen)

<<extend>> is an optional Use Case (may not happen)

Use Case Diagram





What ‘functionality’ (doing words) can you pick out?

<> Code ⚡ Issues 🛡 Pull requests ⏴ Actions 📁 Projects 📖 Wiki 🛡 Security 🛠 Insights ⚙ Se

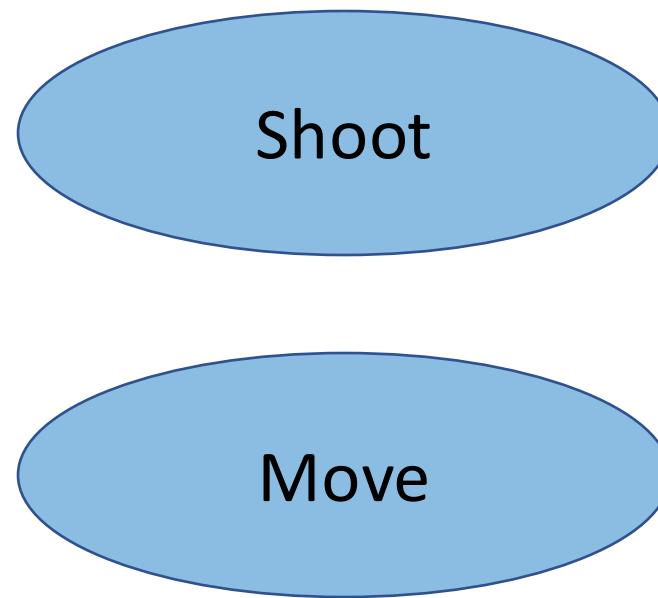
Space Invaders

Nicholas Day edited this page now · 1 revision

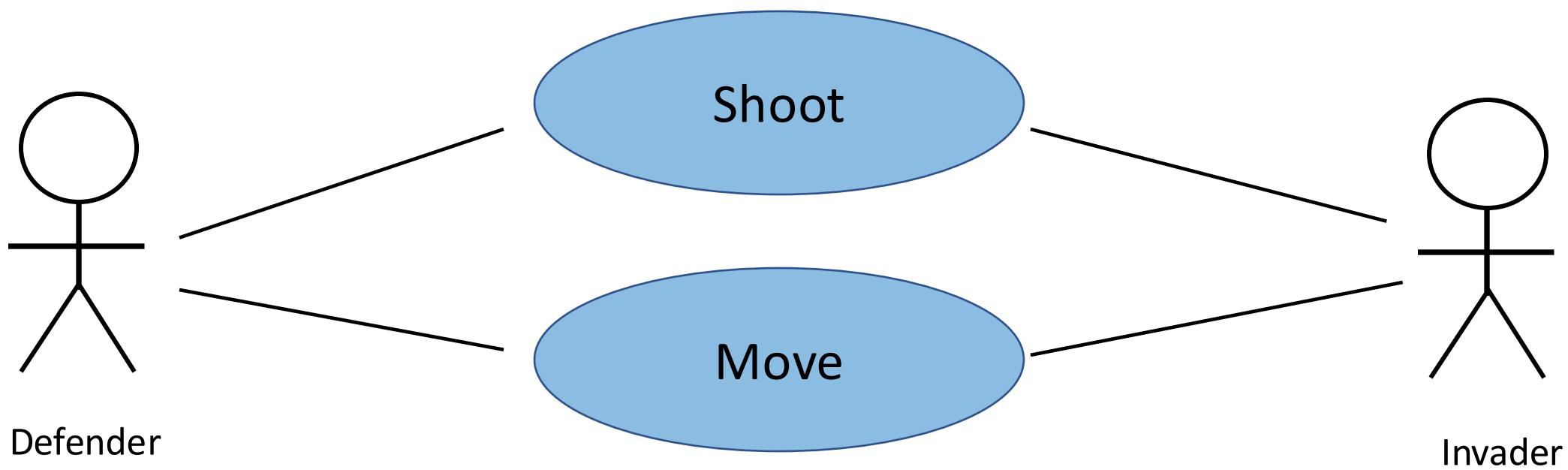
Requirements

- The Defender should move left and right and be able to fire bullets
- Invaders must alternate direction as they move down the screen
- Invaders speed of travel should increase as fewer remain and levels progress
- Shields should crumble as they shot by both the Invader and the Defender

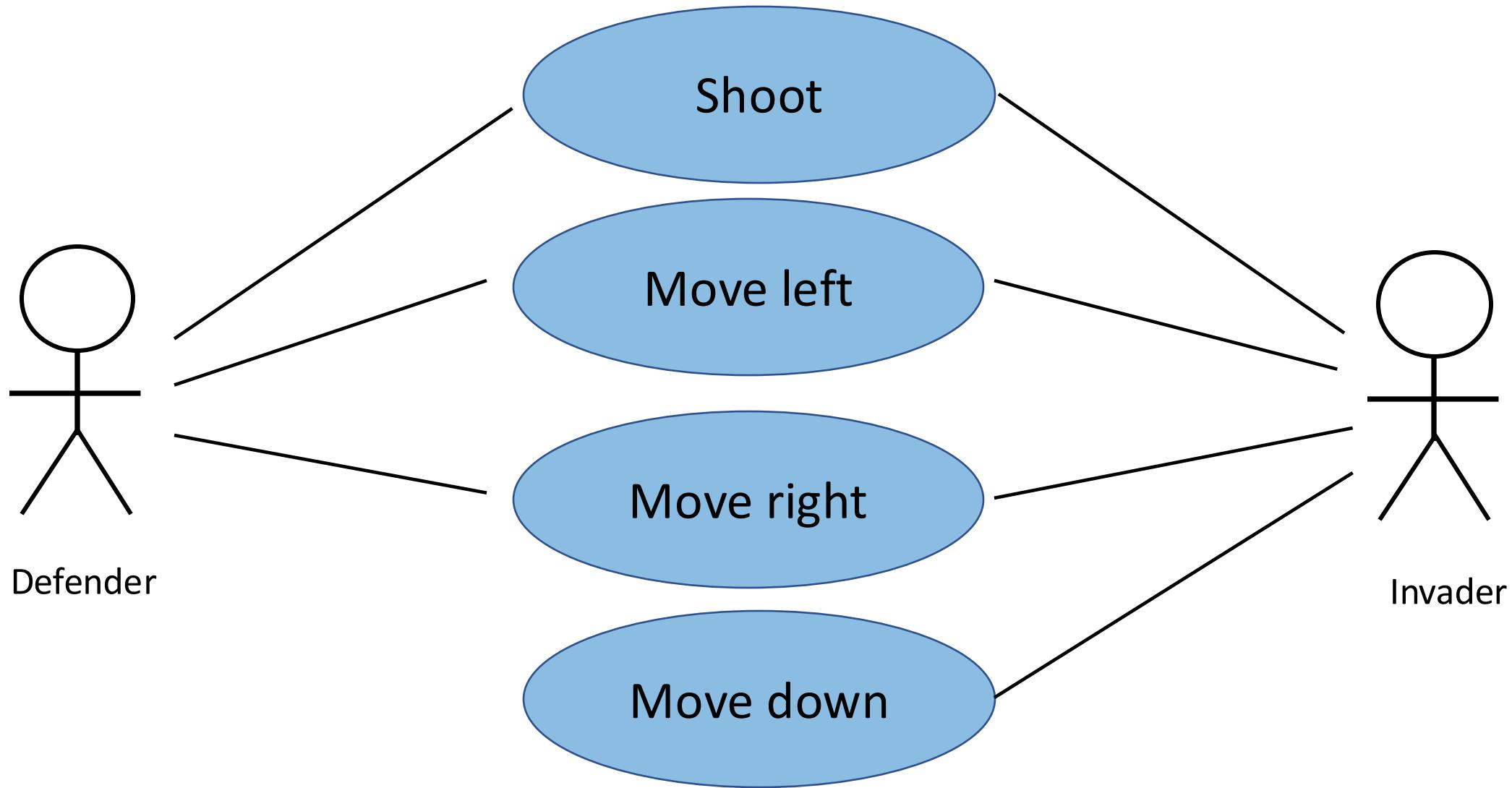
Use Case Diagram - Starting with functions



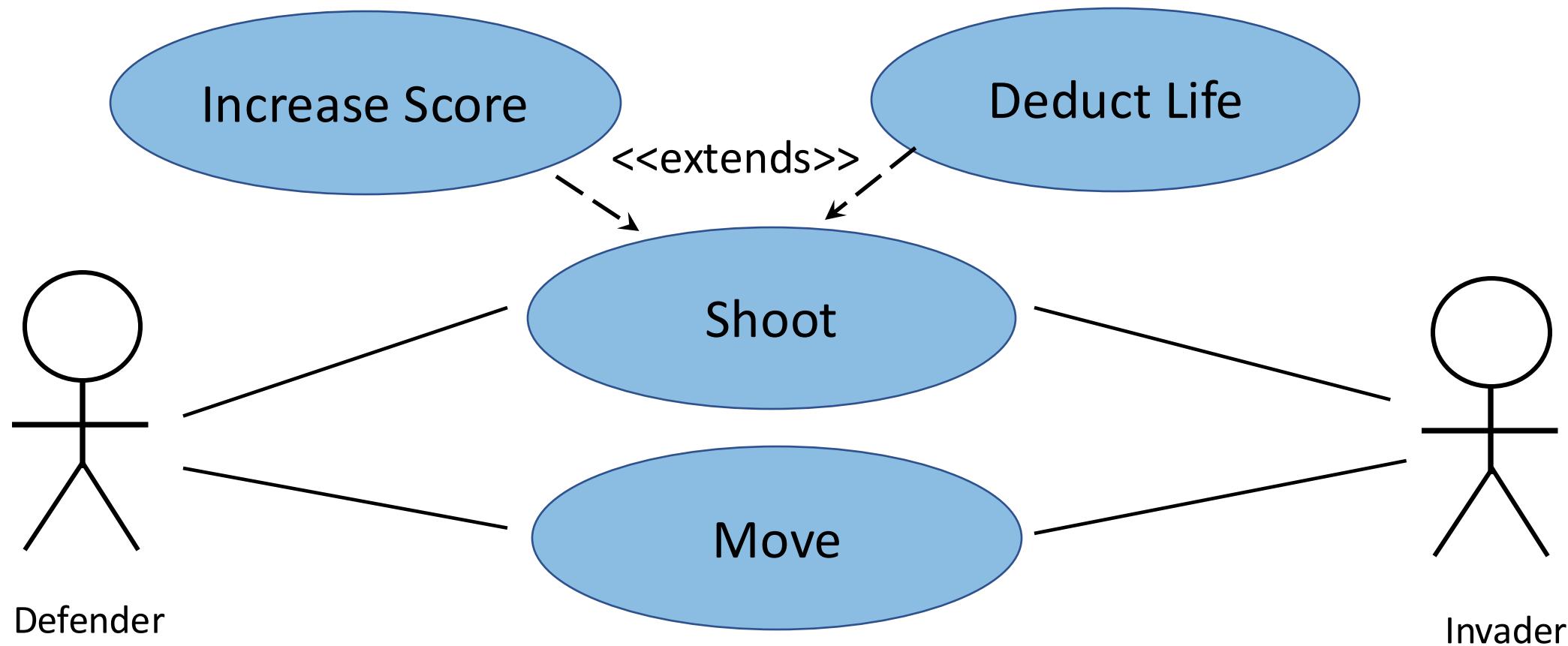
Use Case Diagram – Adding ‘Actors’



Use Case Diagram – Specialist movement



Use Case Diagram – Extension actions

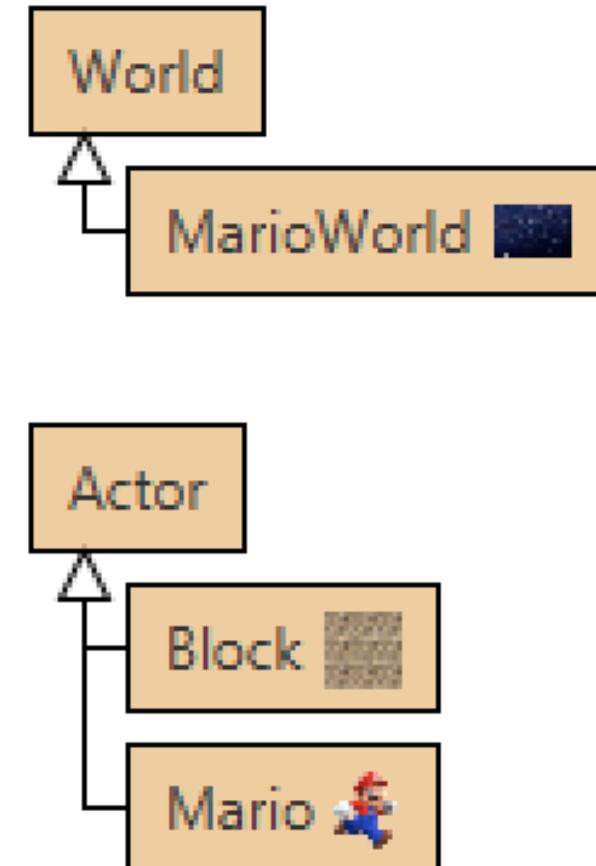
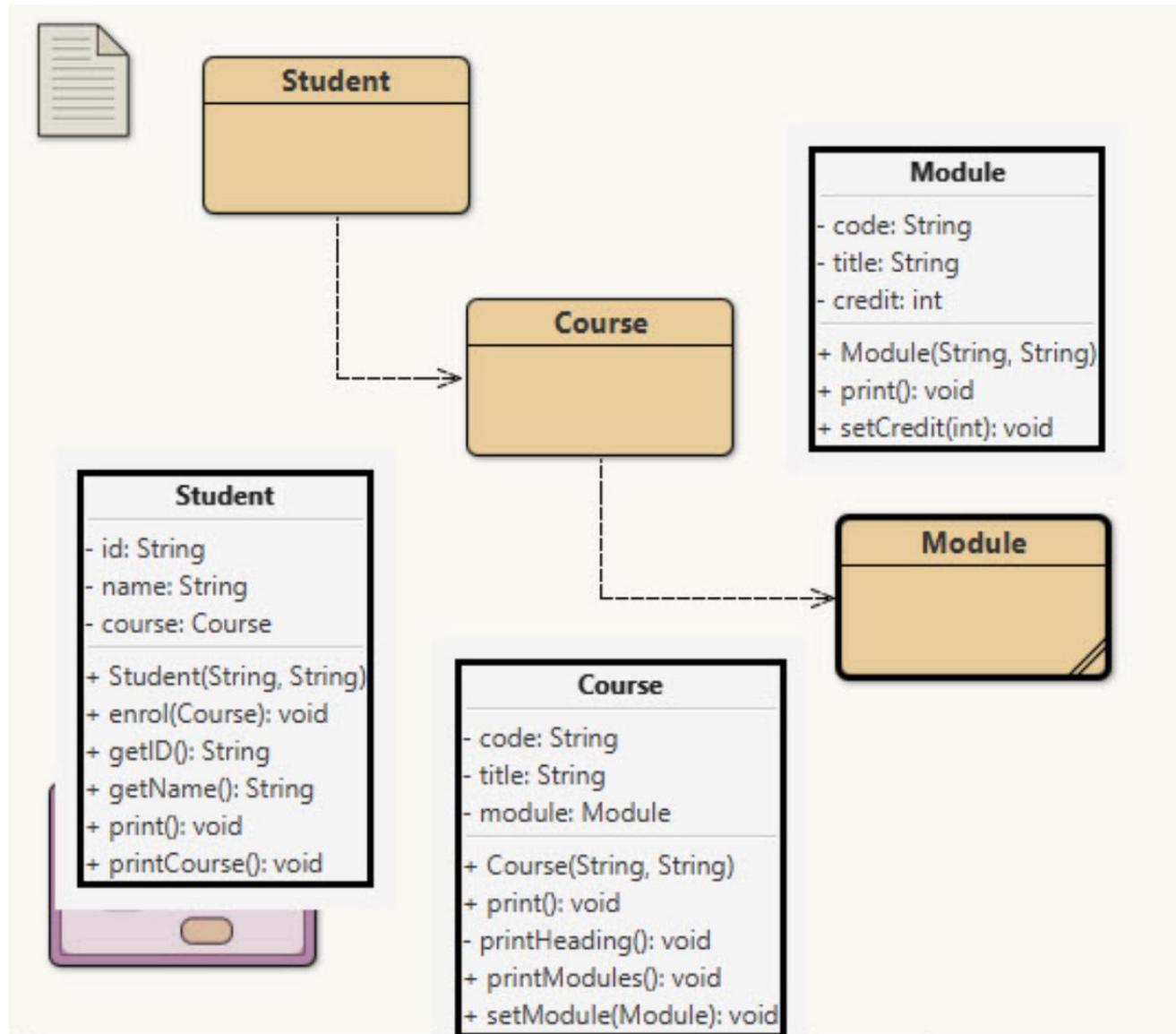


Class Diagrams

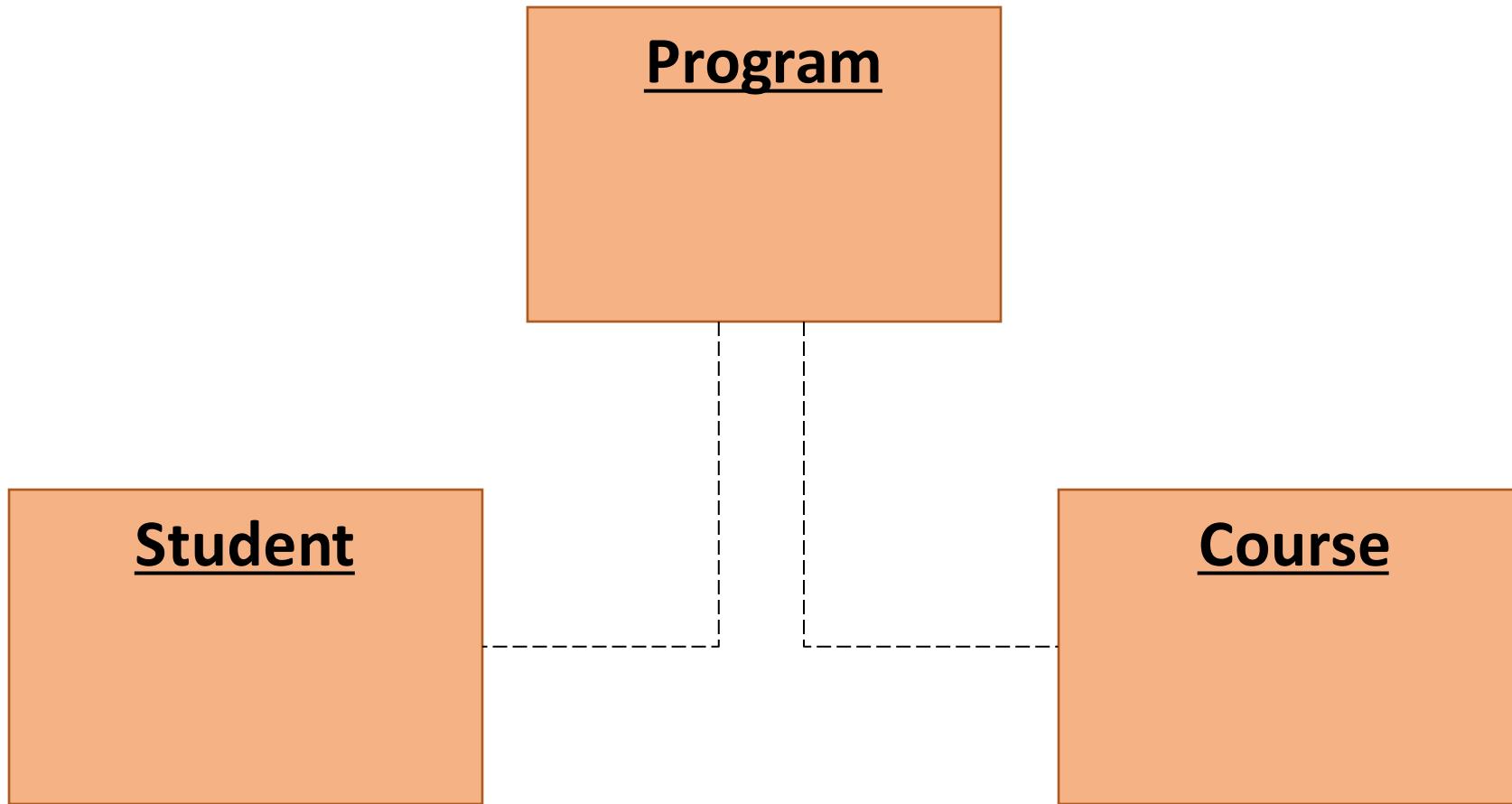
Class Diagrams

- The BlueJ/Greenfoot interface provides a basic version of a Class Diagram because it shows the classes of an application and any associations between them.
- A more detailed Class Diagram would show the **variables** and **methods** written in each class.

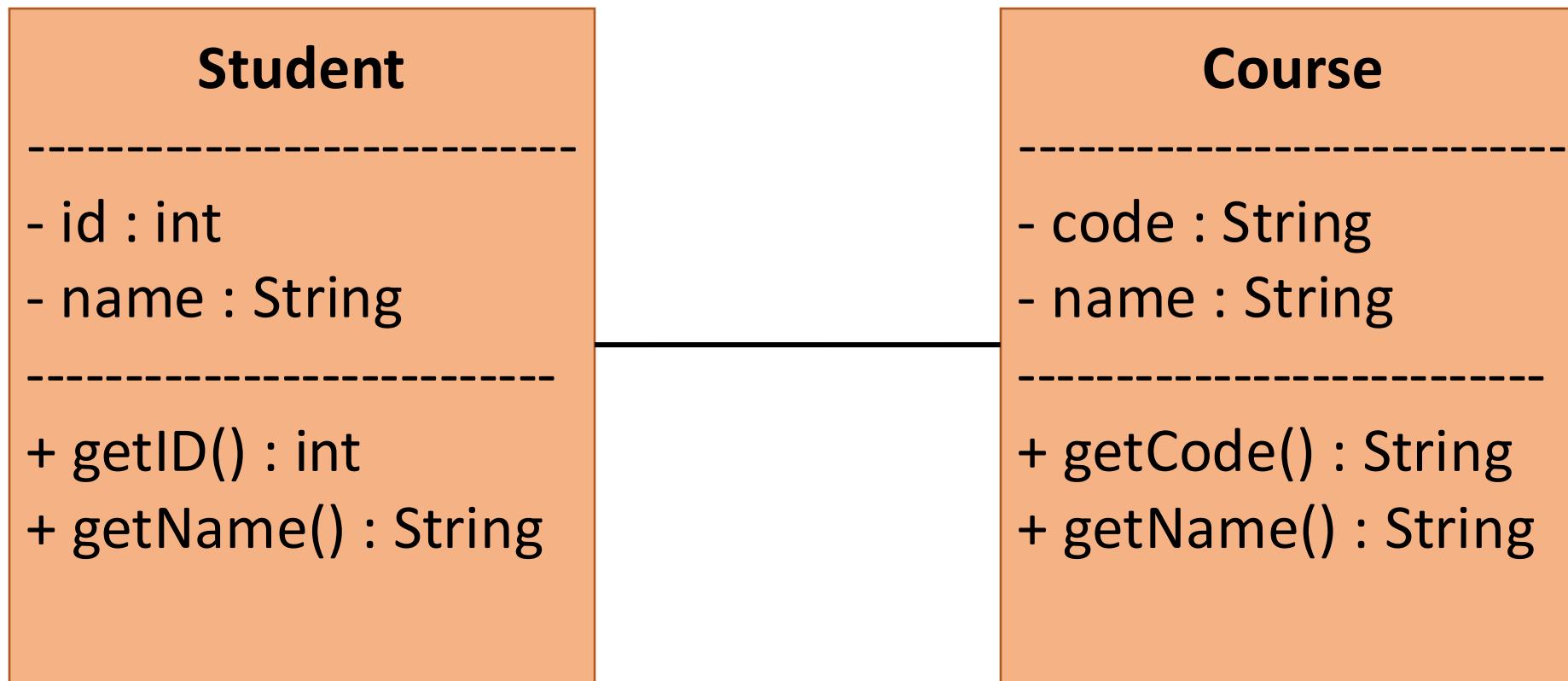
BlueJ / Greenfoot

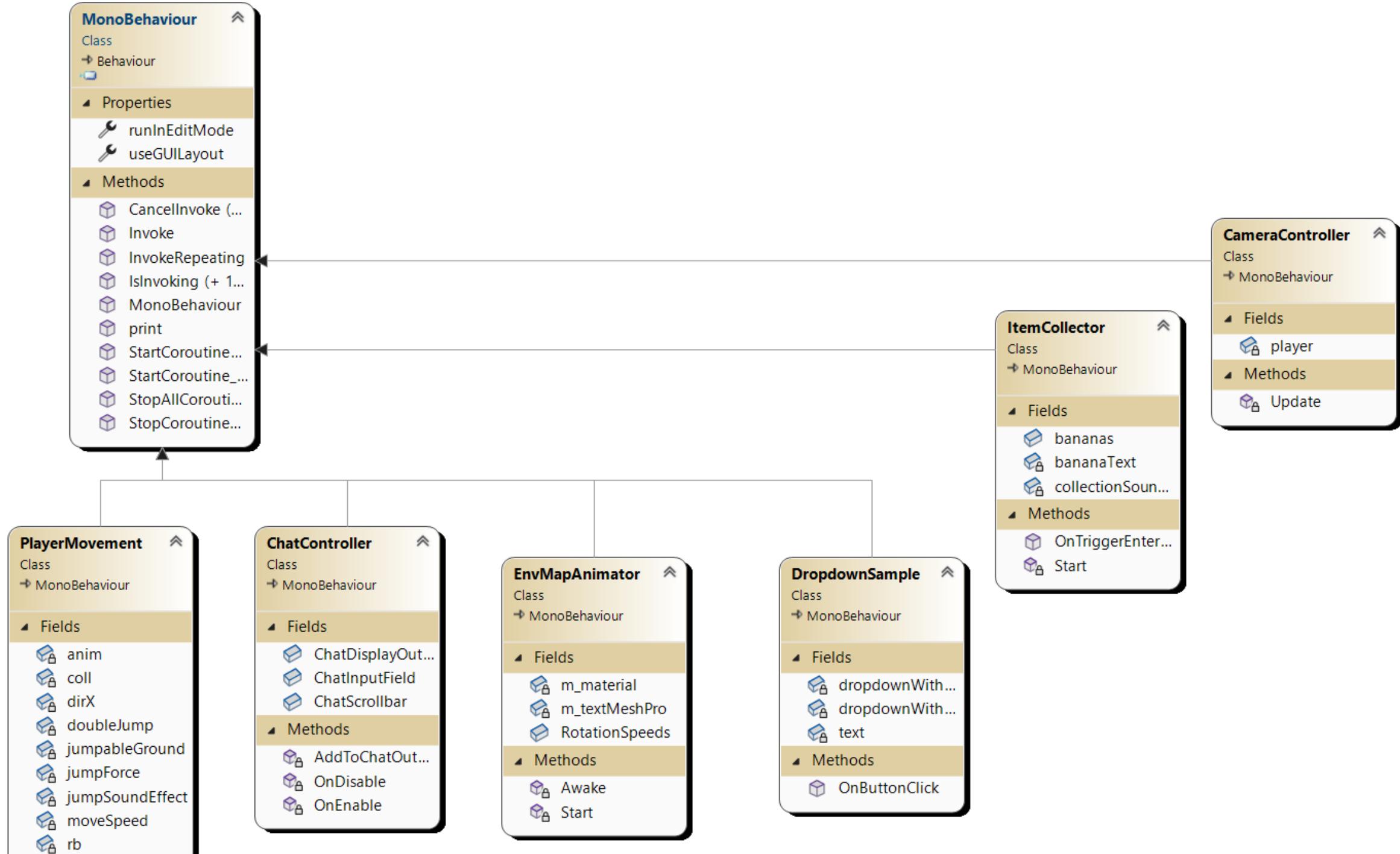


Class View in BlueJ/Greenfoot



UML Class Diagram





How many classes can you pick out?

<> Code ⚙ Issues 🛡 Pull requests ⏴ Actions 📁 Projects 📖 Wiki 🛡 Security 🛠 Insights ⚙ Se

Space Invaders

Nicholas Day edited this page now · 1 revision

Requirements

- The Defender should move left and right and be able to fire bullets
- Invaders must alternate direction as they move down the screen
- Invaders speed of travel should increase as fewer remain and levels progress
- Shields should crumble as they shot by both the Invader and the Defender

Class View – What Entities are there?

Defender

Invader

Class View – attributes?

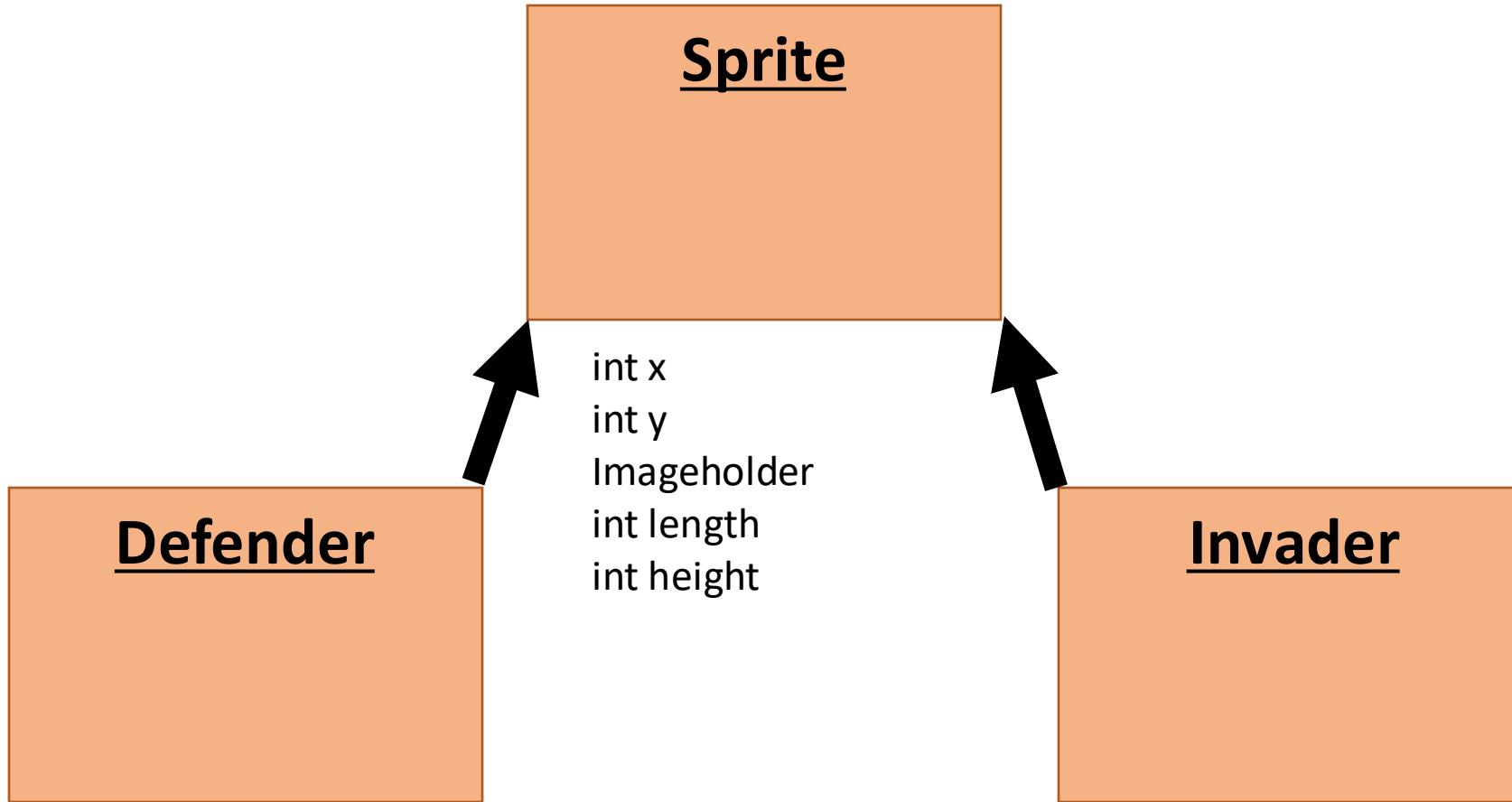
Defender

int x
int y
Imageholder
int length
int height

Invader

int x
int y
Imageholder
int length
int height

Class View – Added a ‘parent’ class



Inheritance?

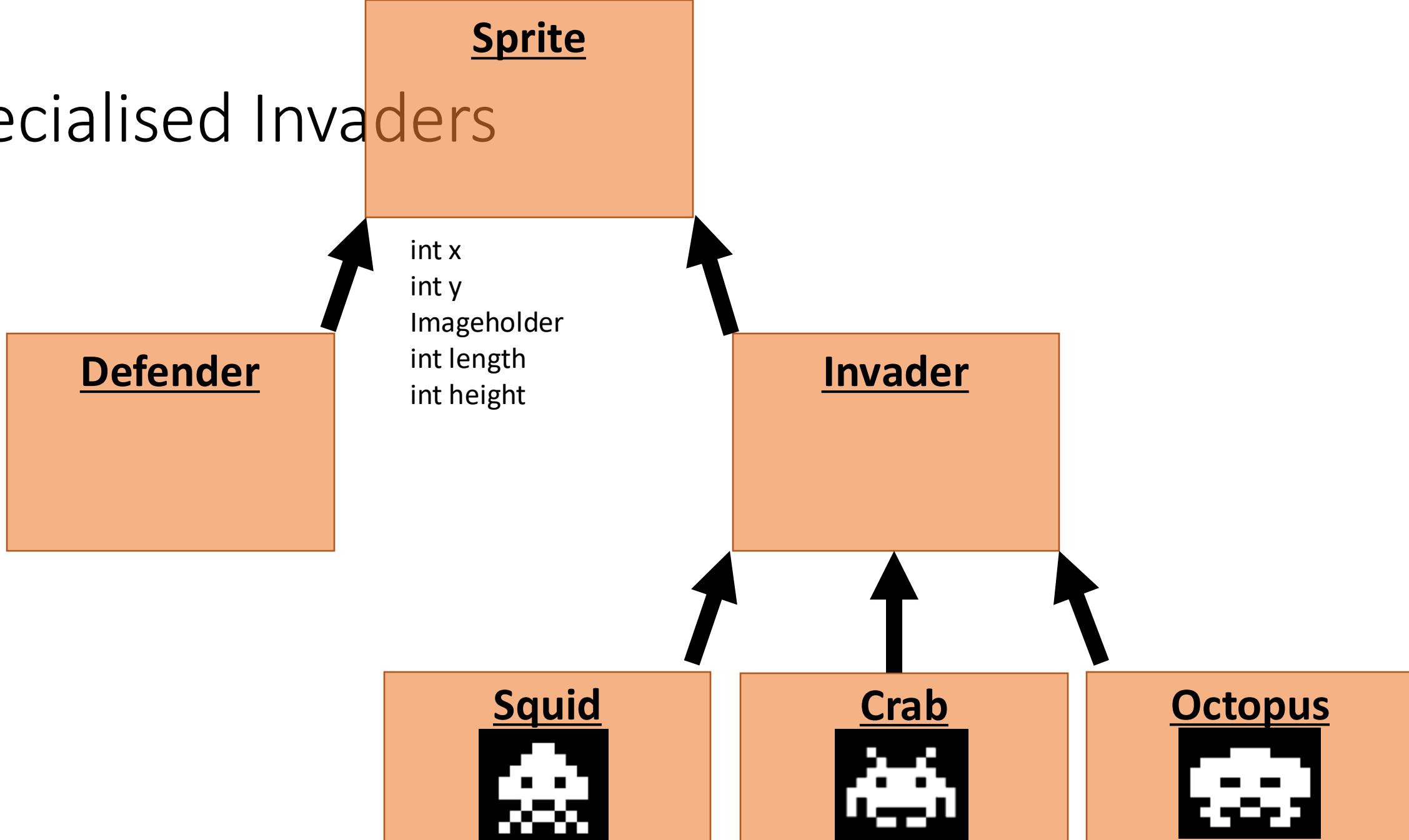
Parent:

Invader

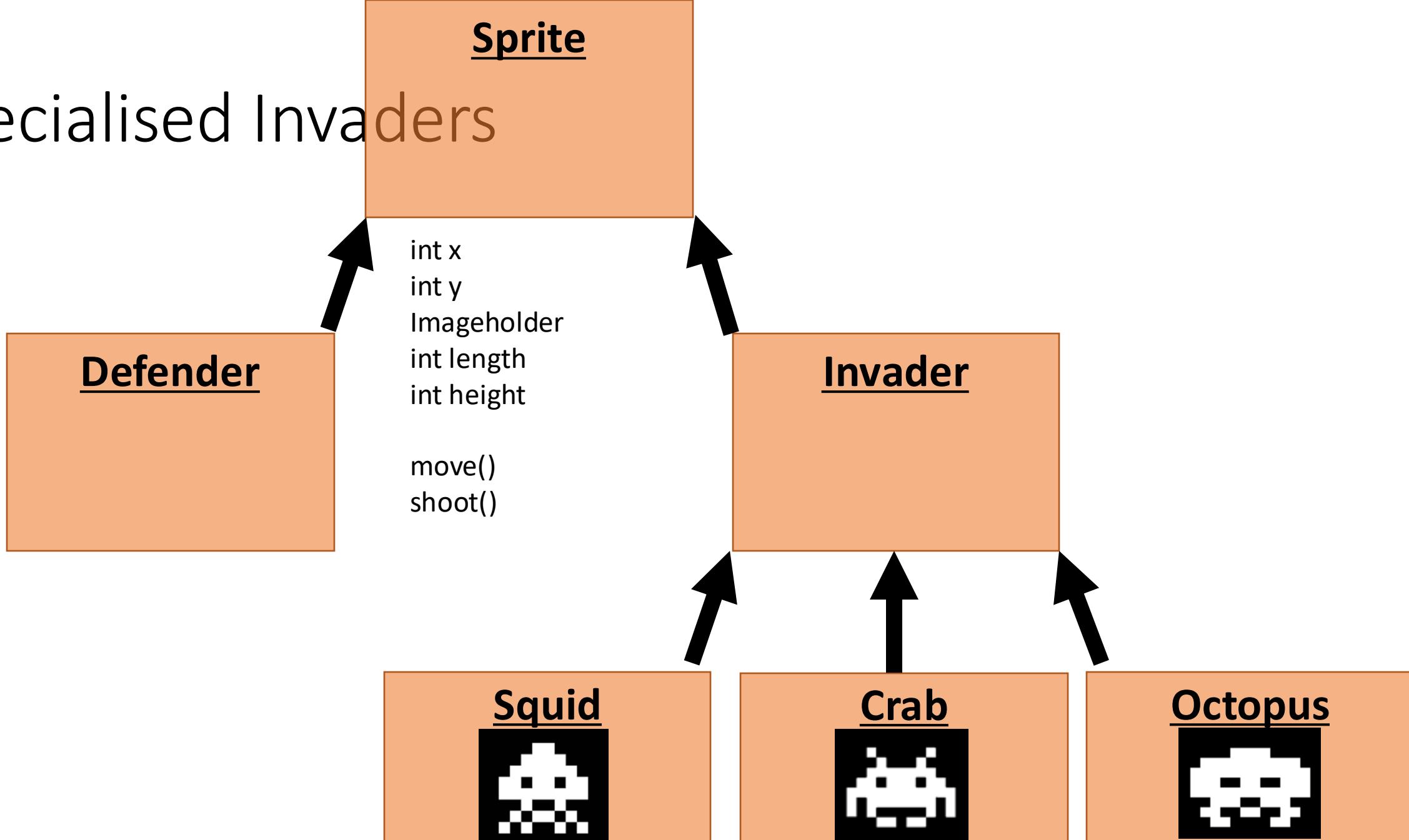
Children:



Specialised Invaders



Specialised Invaders



Sequence Diagram

ERD



User Experience



Make it easy

Easy to use

Good usability, short learning curve and efficiency to perform tasks

The importance of the first contact

It sets the tone of the experience and will dictate whether or not users continue to use the product

There is no single rule

The interface design is directly conditioned by the users who are going to use it and by the context



[Store](#)[Mac](#)[iPad](#)[iPhone](#)[Watch](#)[AirPods](#)[TV & Home](#)[Entertainment](#)[Accessories](#)[Support](#)

iPhone 15

New camera. New design. Newphoria.

[Learn more >](#) [Buy >](#)

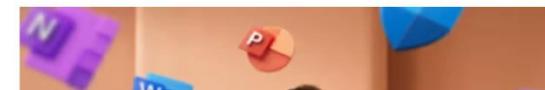


Surface Laptop Studio 2

Save up to 10% student discount on Surface Laptop Studio 2 - the ultimate powerhouse all-in-one laptop.

[Shop Surface Laptop Studio 2](#)[Shop all deals >](#)

① < ● ○ >

[Choose your Microsoft 365](#)[Shop for Surface devices](#)[Buy Xbox games and consoles](#)[Shop for students](#)[Shop for accessories](#)[Shop Business](#)

Dec. 31, 1935.

C. B. DARROW

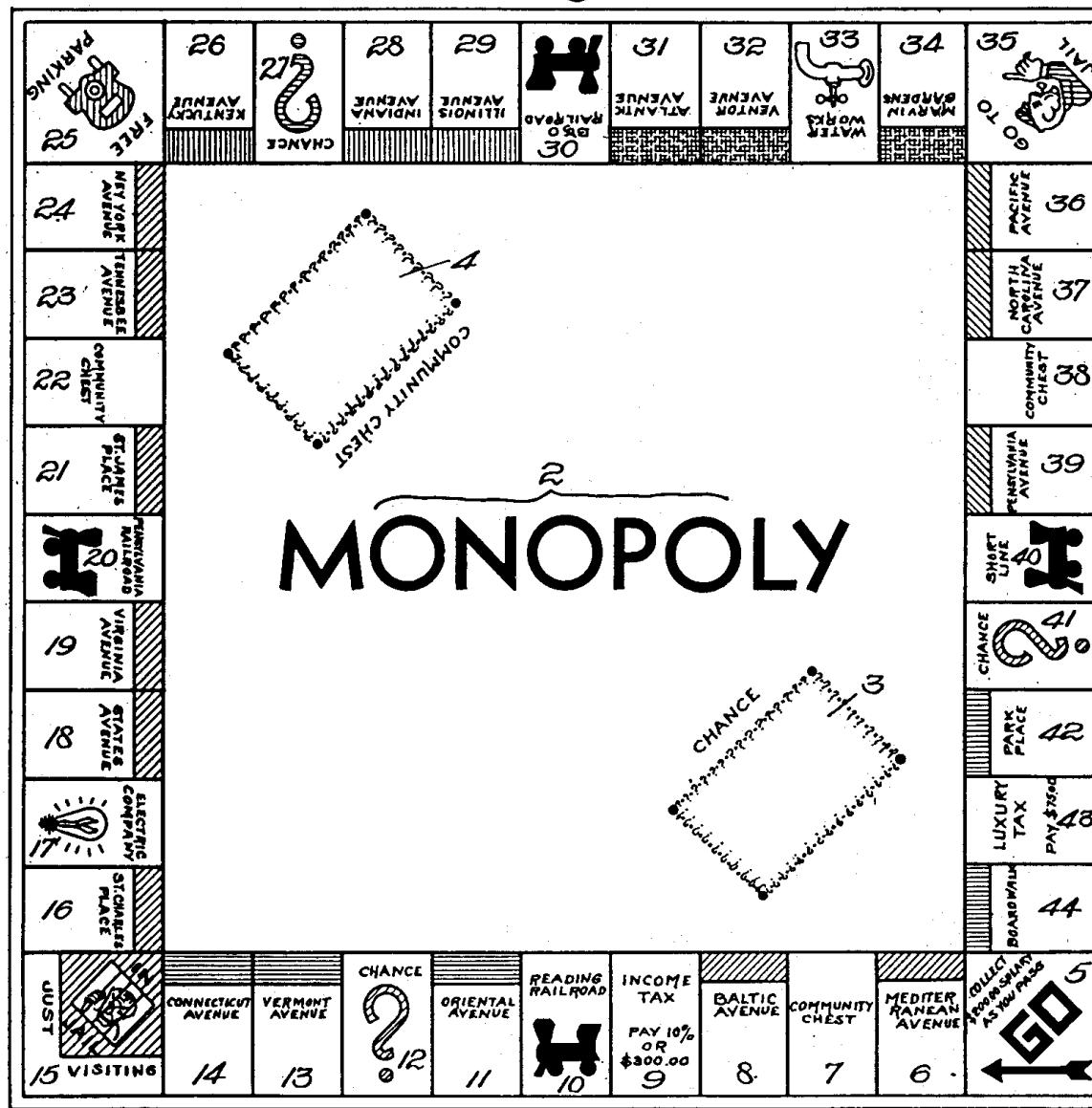
2,026,082

BOARD GAME APPARATUS

Filed Aug. 31, 1935

7 Sheets-Sheet 1

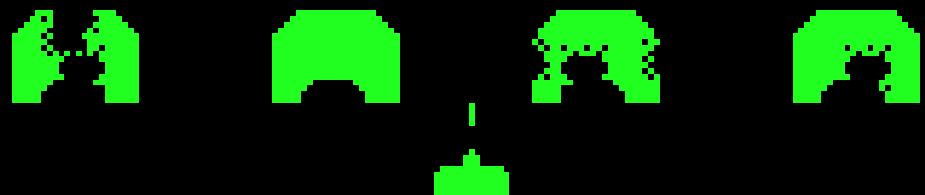
Fig. 1.



SCORE<1> HI-SCORE SCORE<2>

0070

0888



3



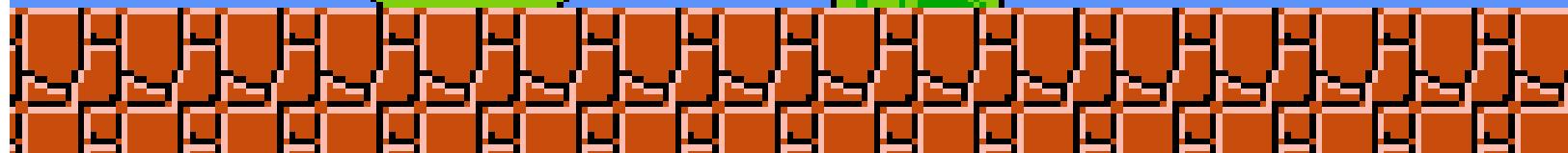
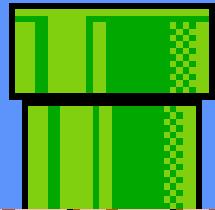
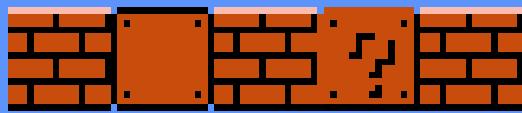
CREDIT 00

MARIO
001300

0x01

WORLD
1-1

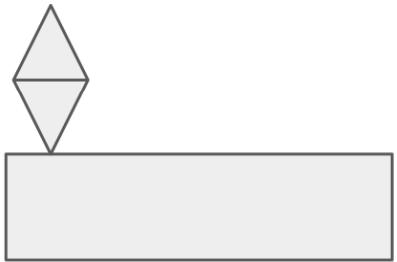
TIME
380



Level Design

Basic mapping.

End

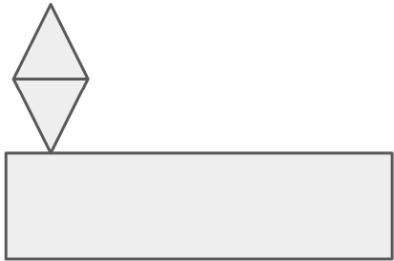


Player

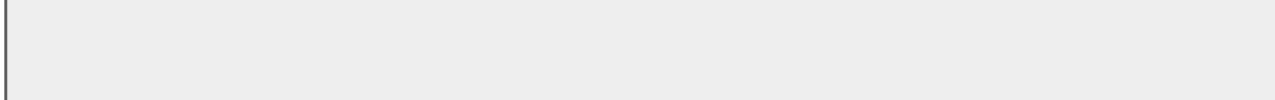


Basic mapping.

End

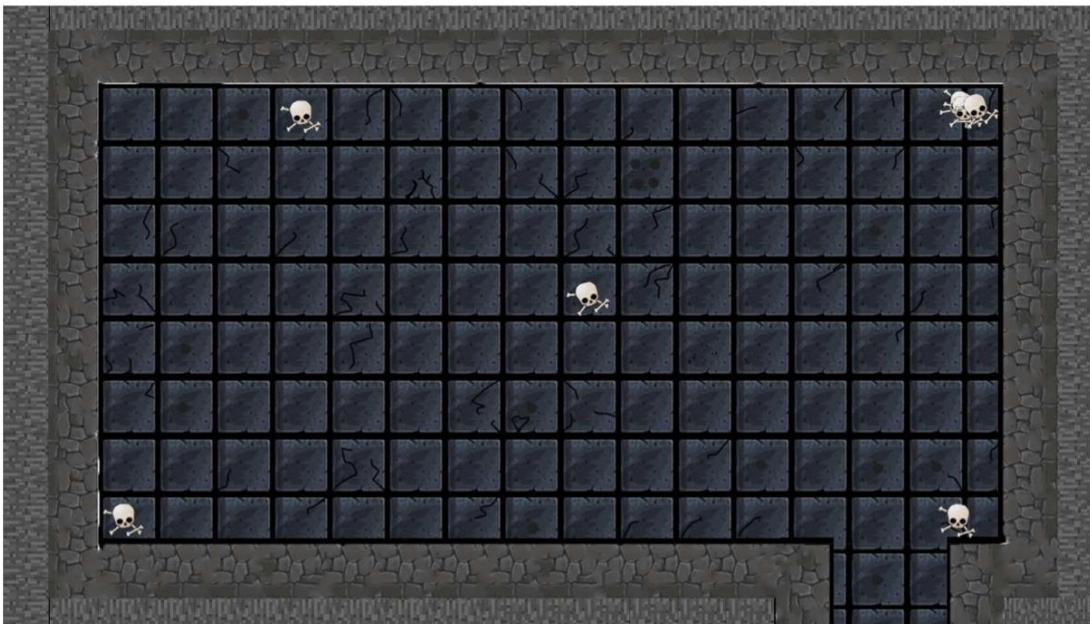


Player

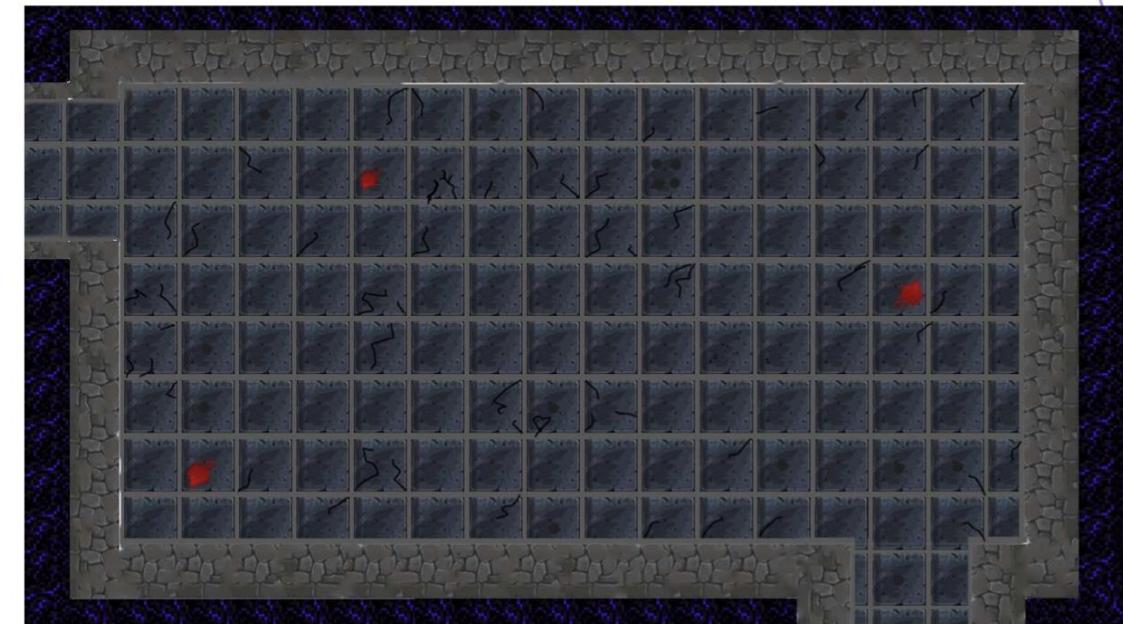


WORLDS / LEVELS

Level 1



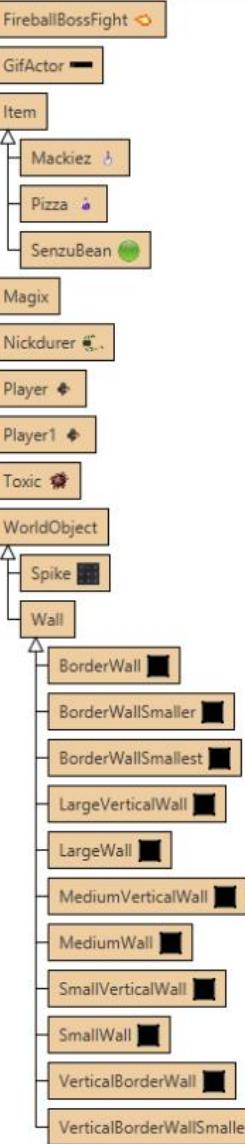
Level 2



World Subclasses

Share...

Magix: 0



BG1	2000H	8
BG2	2400H	
BG3	2800H	1 0 3 2
BG4	2C00H	
		5 4 7 6

BG PLANNING SHEET

PHASE

年月日
テ
サ
イ
ナ

MEMO: W1-2

② ①

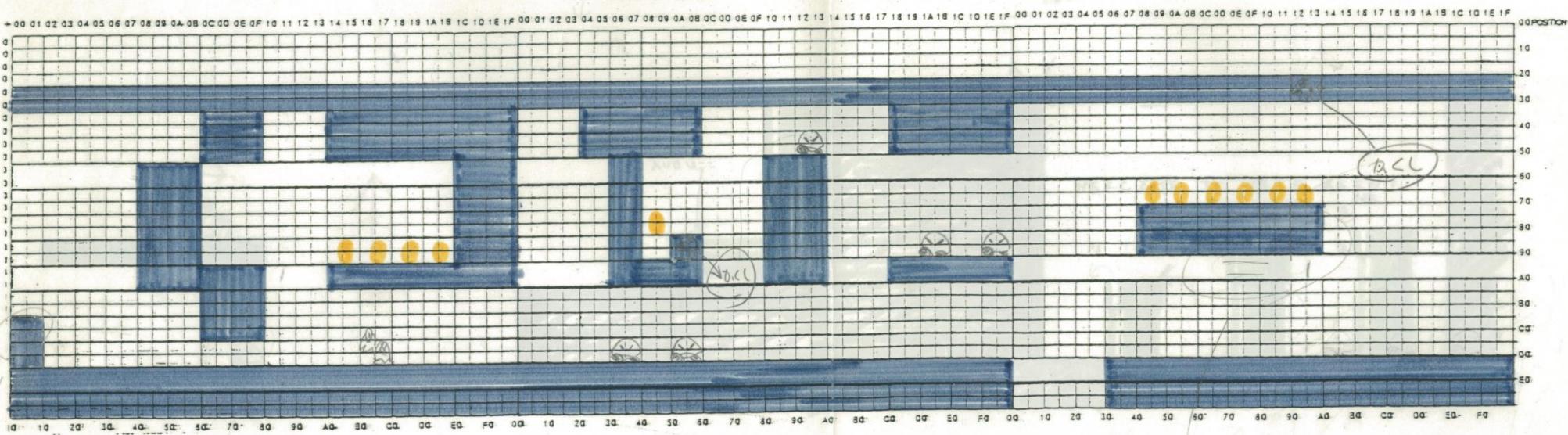
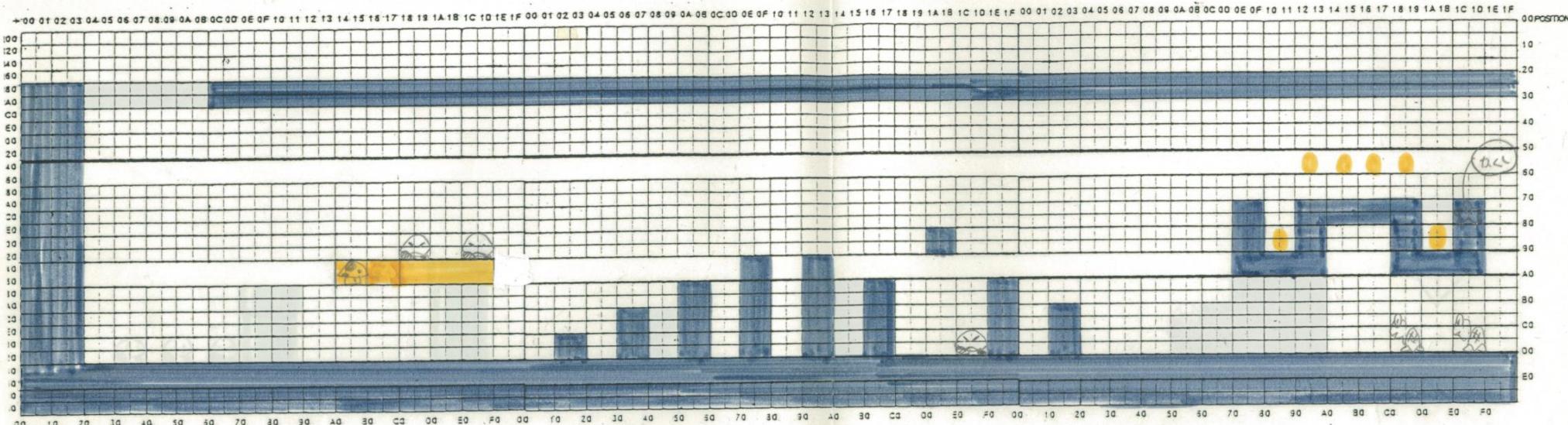
150

AM
PM

時 分
ナ

ノ
ド
ウ
ノ
ド
ウ
ノ
ド
ウ

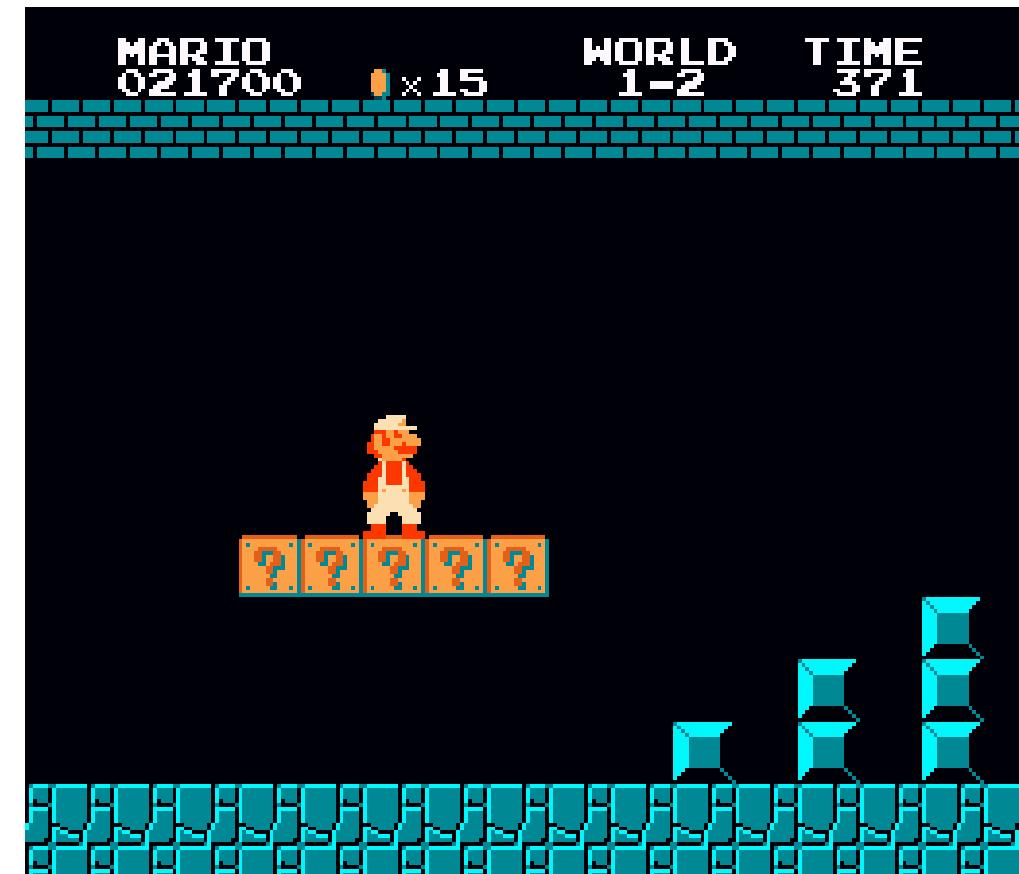
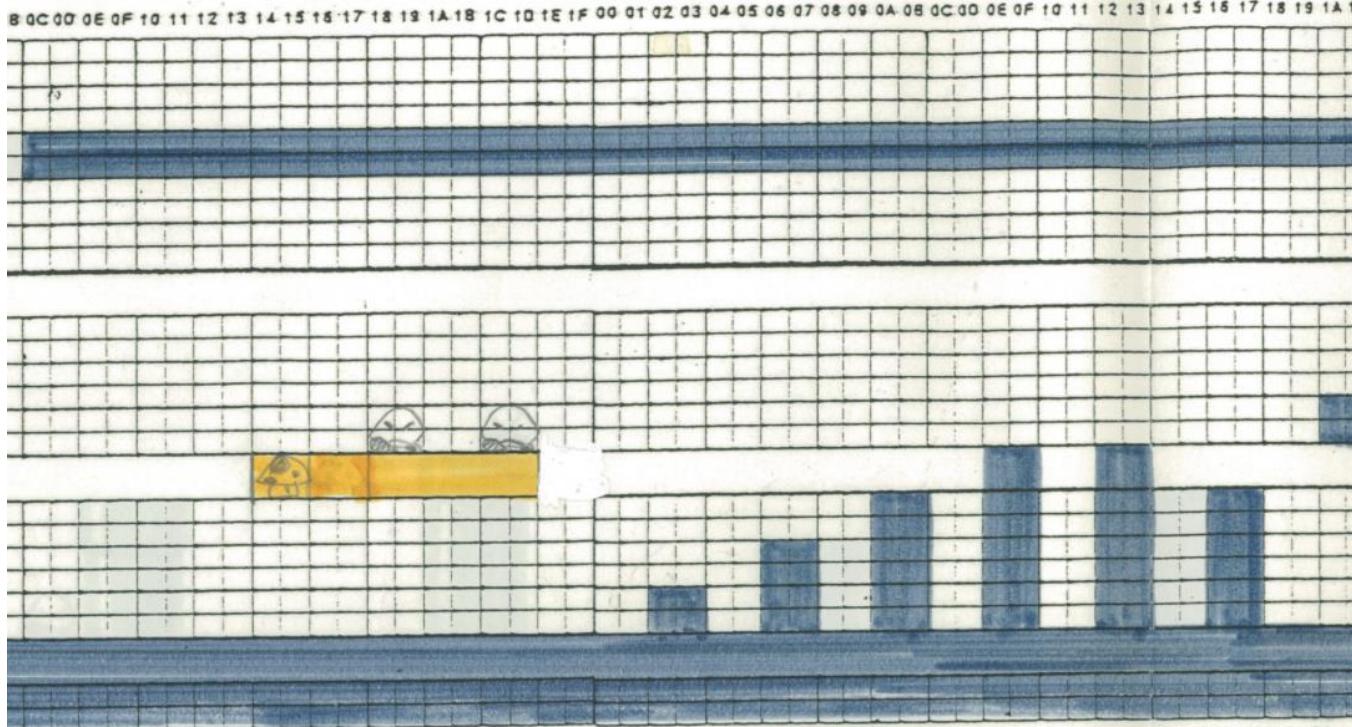
< Under ground - Q >

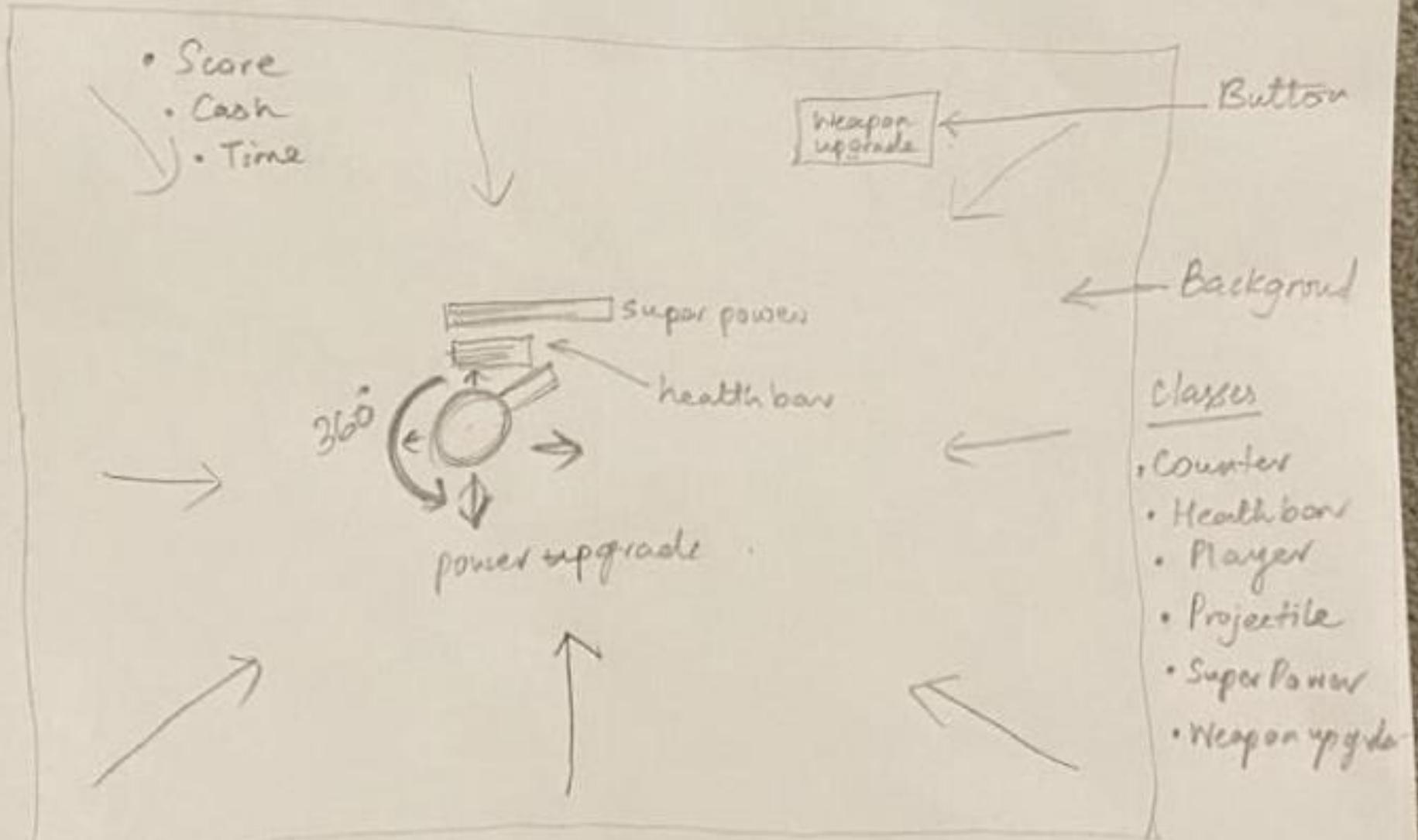


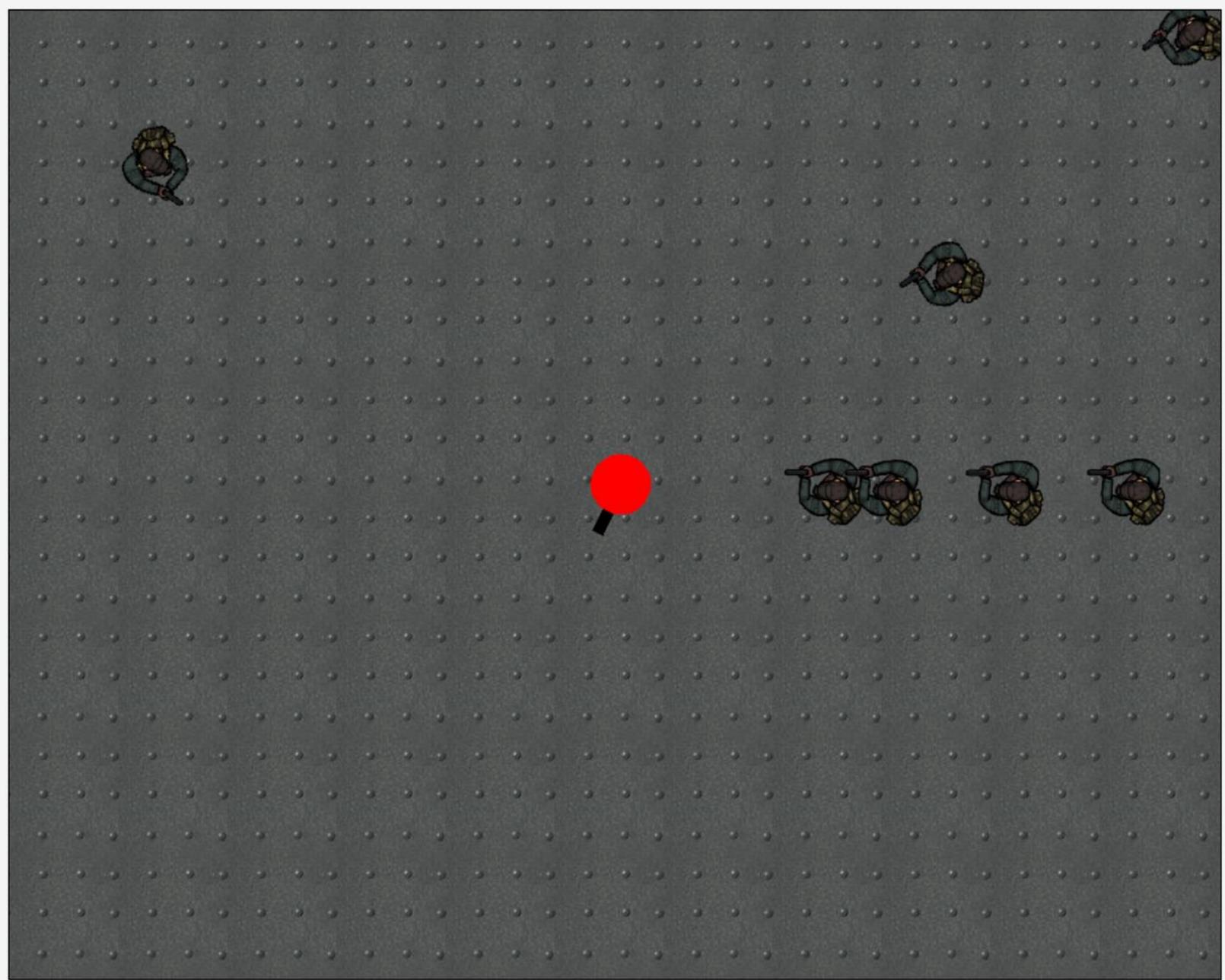
BG1	2000H	8
BG2	2400H	
BG3	2800H	T
BG4	2C00H	
5	4	7
6		

BG PLANNING SHEET

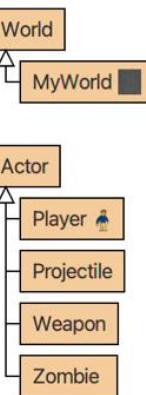
MEMO: W1 - 2 ① 150







Share...



> Act

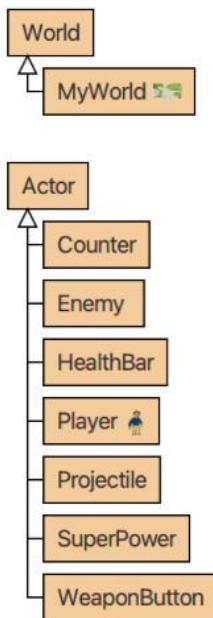
Run

Reset

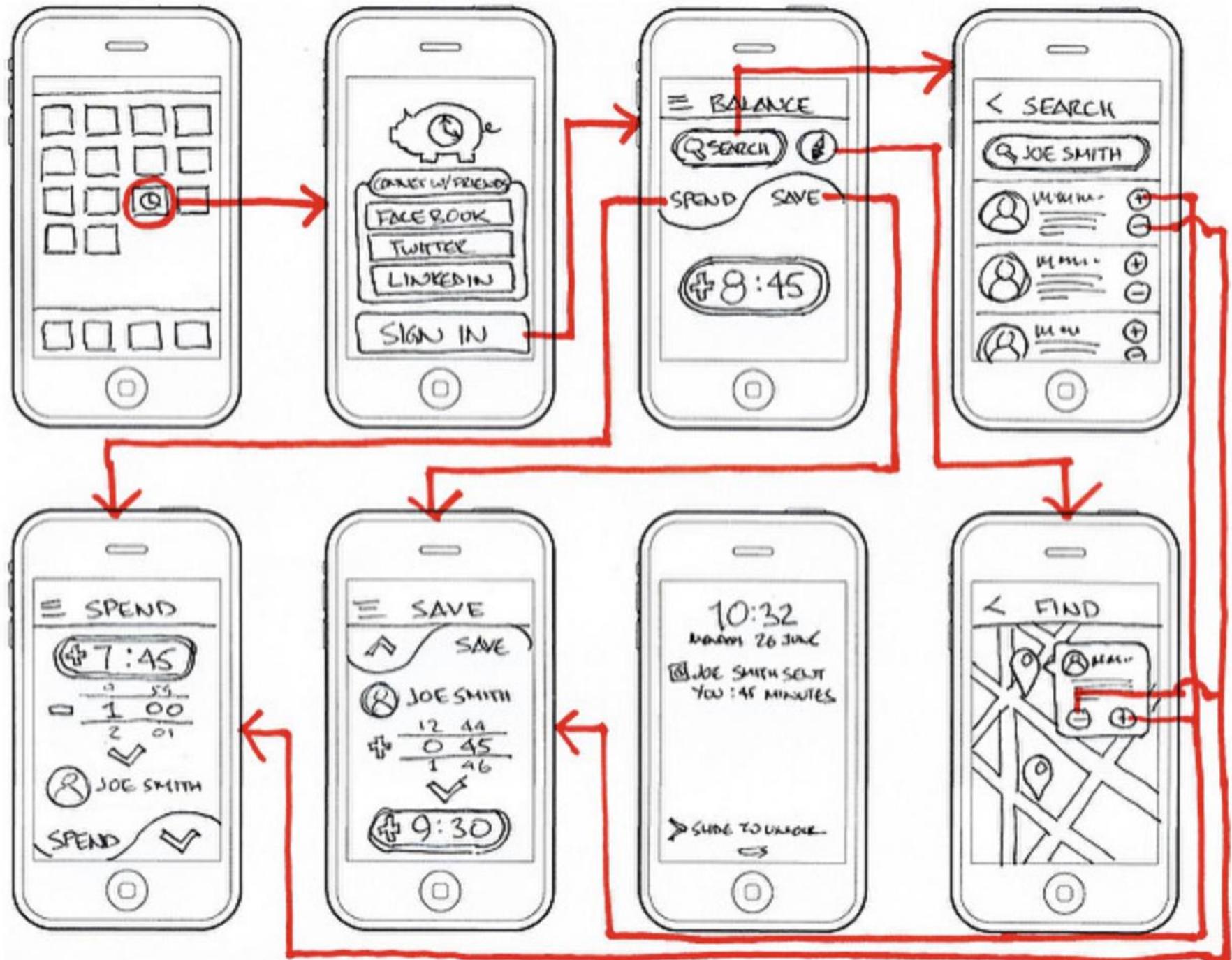
Speed:

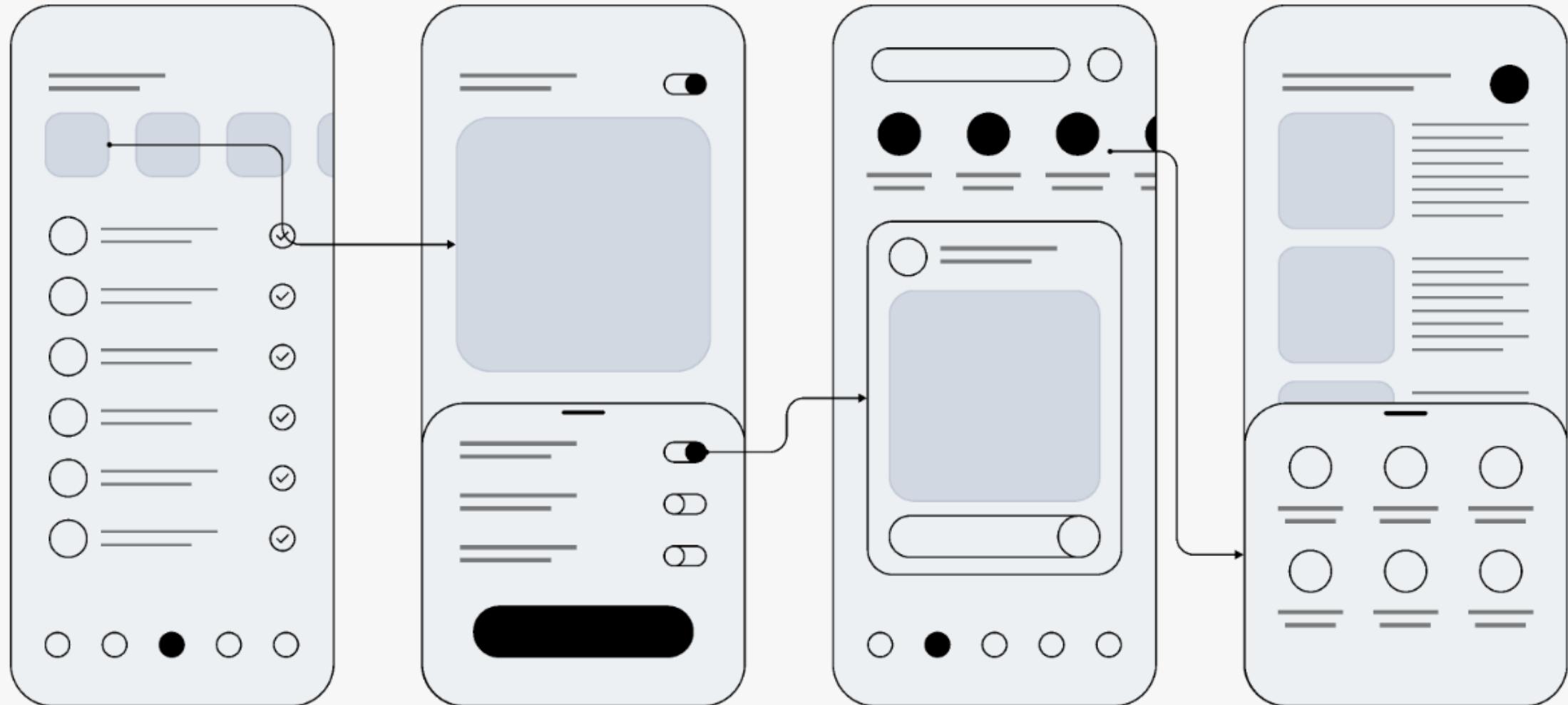


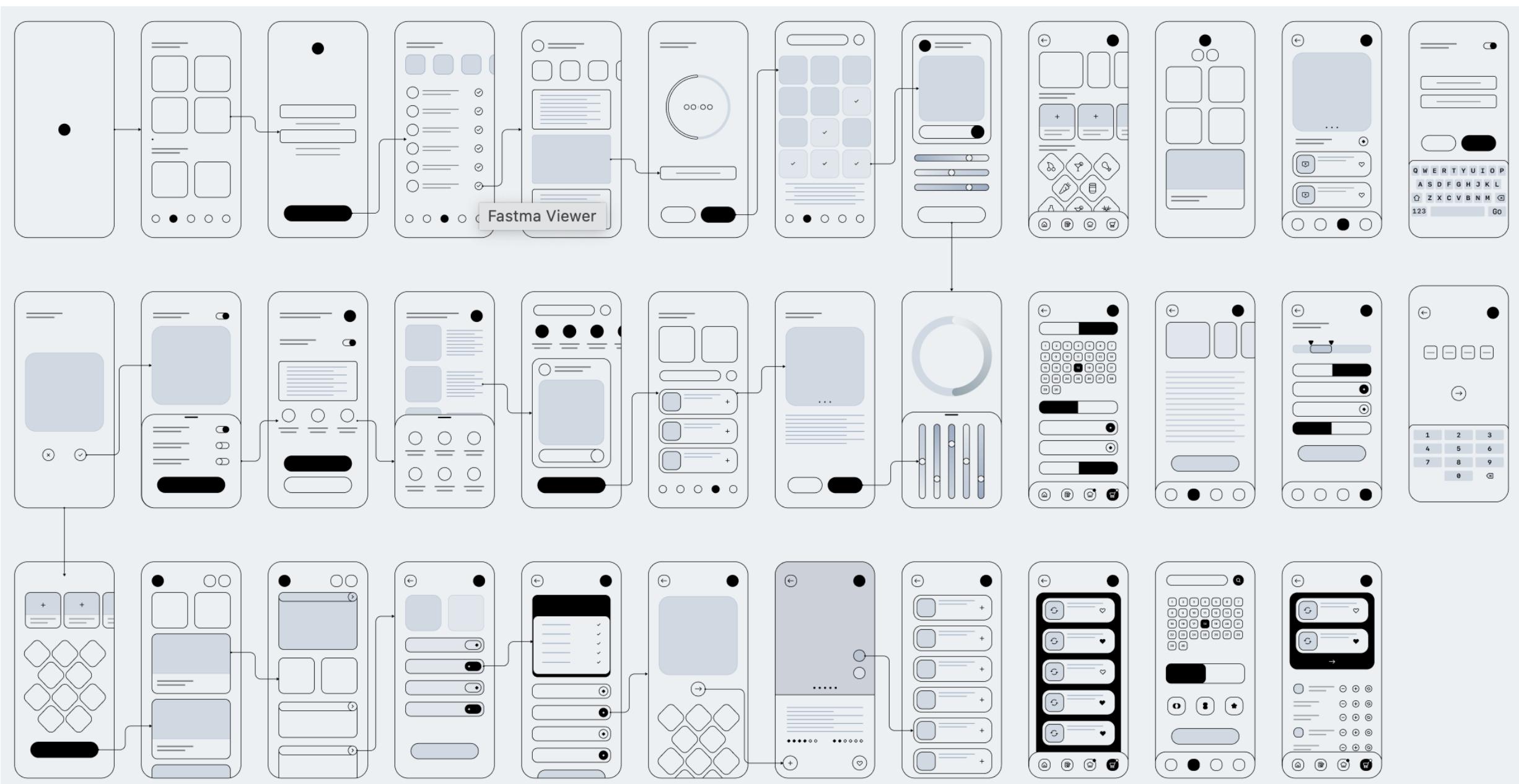
Share...



Wireframing







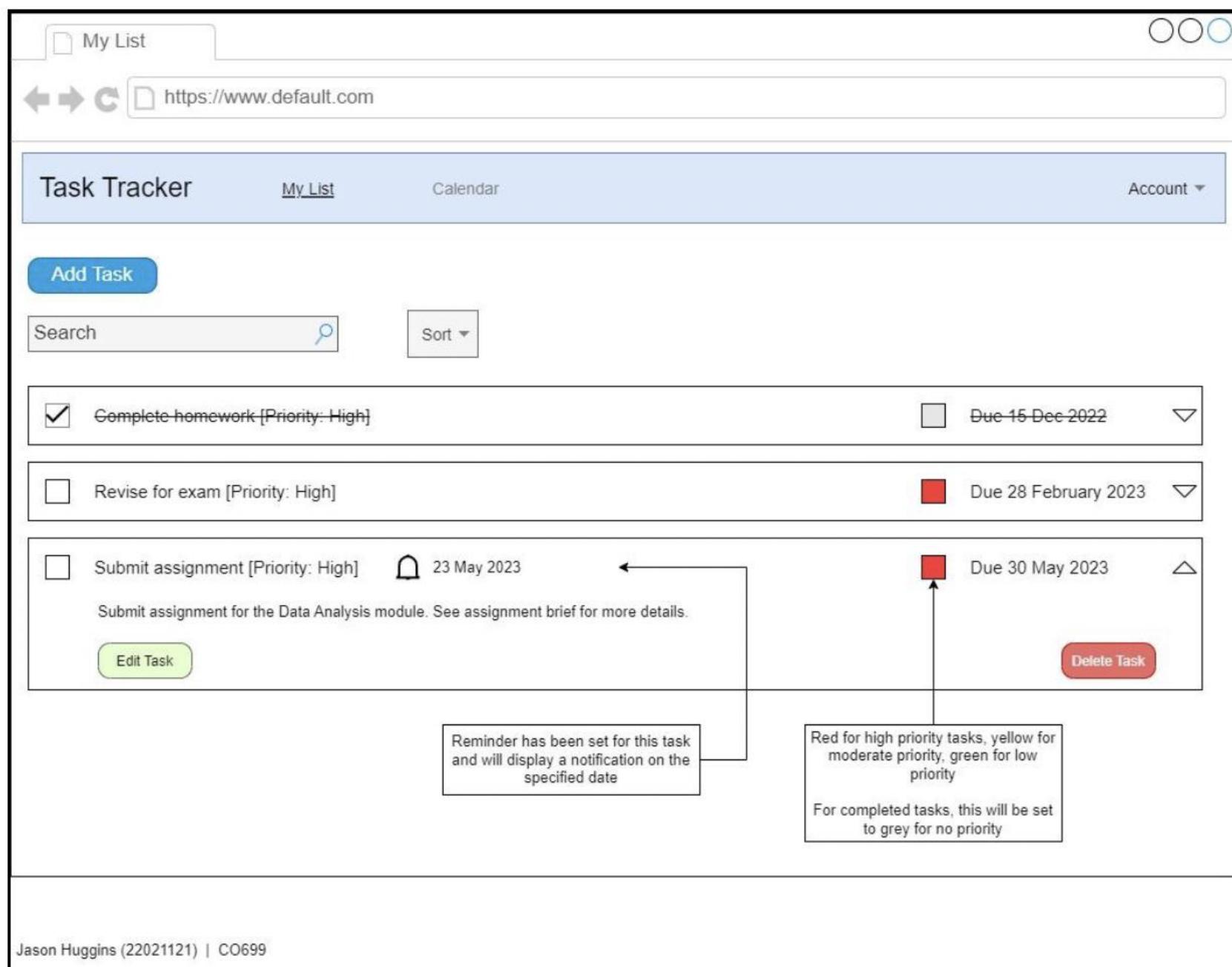


Figure 8 – “My List” Page [Sample Colour Scheme; Refined Wireframe]

The screenshot shows a web browser window titled "My List - Task Tracker". The URL in the address bar is "localhost:44353/UserTasks/Index". The page header includes links for "Task Tracker", "My List", and "Calendar". Below the header, the main content area has a title "My List". It features a blue "Add Task" button, a search bar with a "Search" input field and a "Submit" button, and a table listing tasks. The first task in the table is "Complete assignment" with a due date of "Due 15/02/2023", priority "High", and status "Completed". It includes "Edit" and "Delete" buttons. At the bottom left, there is a link to "Privacy". The footer contains the text "Jason Huggins (22021121) | CO699 Project".

<input type="checkbox"/> Completed	Due 15/02/2023
Complete assignment See assignment brief for more details.	Priority: High

[Privacy](#)

Jason Huggins (22021121) | CO699 Project

Figure 26 - Current Design for “My List” Page

Storyboarding

Guest Story



Host Story

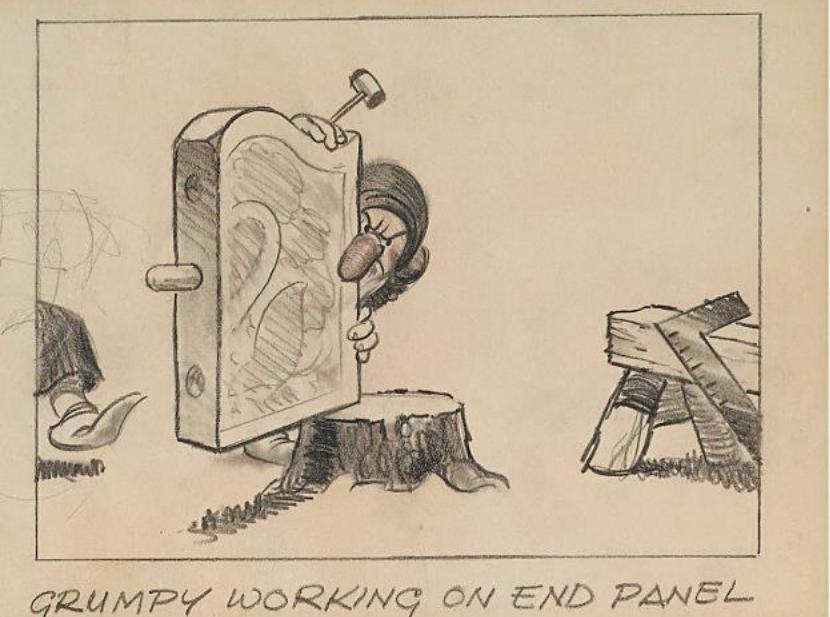


QUICK STUDIES

How Airbnb Used Storyboards to Understand the Customer Experience

Literally seeing how your customer interacts with your organization can provide new insights into better user experiences, business opportunities, and team priorities.

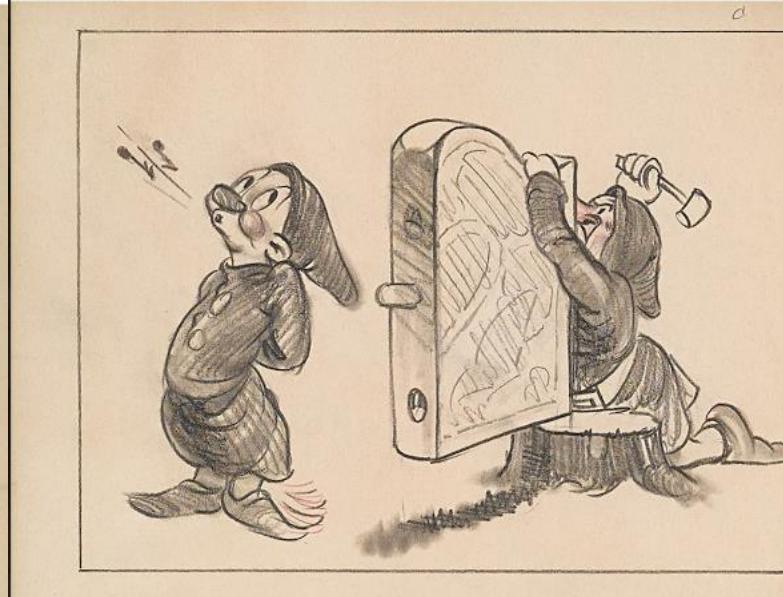
1 MIN READ



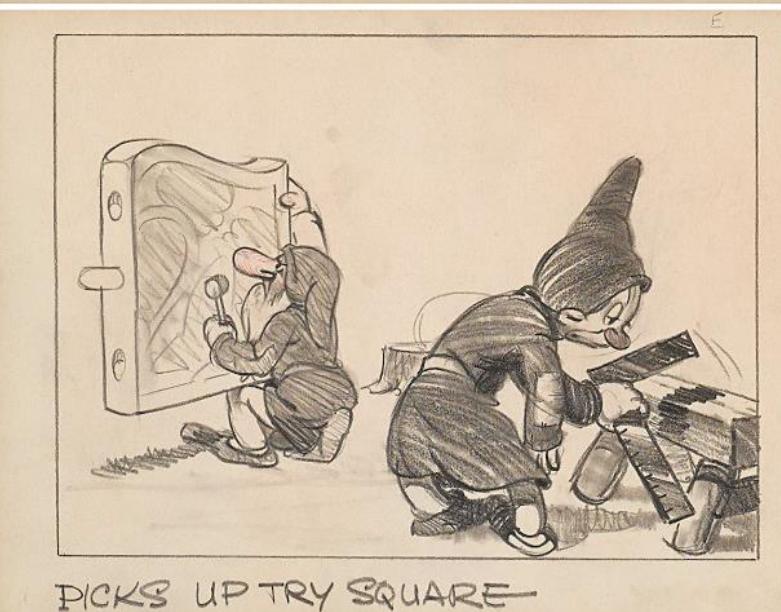
GRUMPY WORKING ON END PANEL



DOPEY MISTAKES PANEL FOR DOOR



IMPATIENT— GOES THROUGH "DOOR"



PICKS UP TRY SQUARE



GOES OUT — SLAMMING 'DOOR'

A few years back, Brian Chesky, CEO of Airbnb was reading the biography of Walt Disney, and was inspired by Disney's creation of a storyboard to produce his first feature film, Snow White. Airbnb then created their own storyboards, imagining and illustrating what their guests and hosts experience while using their service. The practice has two benefits:

- 1 **It's a visual way to better understand your customers' experiences.** As Chesky notes, "As opposed to working out of a spreadsheet or a Google Doc, this is us creating characters and starting to understand the personality of these characters..."
- 2 **It allows you to imagine even the smallest details.** By mapping out your customer's journey, you can better identify potential pain points and anticipate their emerging needs.

These storyboards identified a missing link between Airbnb's website and the real-life customer experience. As a result, Airbnb shifted their focus to optimizing their now widely utilized mobile platform.

STORYBOARD



- MAKES NOTE OF SUPPLIES NEEDED ON CLIPBOARD
- PHYSICAL INVENTORY

PERSONA:

CORPORATE BUYER,
JAMES



- SELECTS ITEMS FROM FAVORITES LIST
- USES DESKTOP & SUPPLY LIST AS TOOL

SCENARIO:

REPLENISH OFFICE SUPPLIES



- RECEIVES SHIPMENT WINDOW W/ ORDER SUBMISSION
- SETS PLAN FOR RESTOCK



Sees commercial



downloads app



Places Order

Hunger continues



estimated pick
up time given in
the app



DRIVES TO
RESTAURANT



Take
Our
SUR

complet
Survey



COUPON
e-Coupon
in inbox



food

PERSONA: James

SCENARIO: Replenish Office Supplies

- ① James takes a physical inventory of the supply closet and notes items needed on clipboard.



Peeved



focused

- ② James creates an order by picking items from Favorites list via his office desktop computer. Checks off items on clipboard.



- ③ James adds the shipment arrival date to calendar from confirmation message.



done!