

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**  
Факультет кібербезпеки, комп'ютерної та програмної інженерії  
Кафедра комп'ютерних інформаційних технологій

**ДОПУСТИТИ ДО ЗАХИСТУ**

Завідувач кафедри

Савченко А.С.

« \_\_\_ » \_\_\_\_\_ 2020 р.

**ДИПЛОМНА РОБОТА**  
**(ПОЯСНЮВАЛЬНА ЗАПИСКА)**  
**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ**  
**"МАГІСТР"**

**Тема:** «Кросплатформна гра для стимулювання розвитку просторового мислення»

**Виконав:** Семенов Нікіта Максимович

**Керівник:** к.т.н., доцент Холявкіна Тетяна Володимирівна

**Нормоконтролер з ЄСКД (ЄСПД):**

Райчев І.Е.

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Спеціальність 122 “Комп'ютерні науки та інформаційні технології”

Спеціалізація «Інформаційні управляючі системи та технології (за галузями)»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Савченко А.С.

“ ” \_\_\_\_\_ 2019р.

**ЗАВДАННЯ**

**на виконання дипломної роботи студента**

Семенова Нікіти Максимовича

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): «Кросплатформна гра для стимулювання розвитку просторового мислення» затверджена наказом ректора №2175/ст. від 14.10.2019р..
2. Термін виконання проекту (роботи): з 14.10.2019р. по 09.02.2020р.
3. Вихідні данні до проекту (роботи): просторове мислення, відеогра.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, аналітичний огляд і постановка завдання, висновок.
5. Перелік обов'язкового графічного матеріалу: графічний інтерфейс користувача, приклади рівнів, приклади схожих ігор.

### **КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1.	Проаналізувати літературу та джерела за темою дипломного проекту.	13.10.19– 20.10.19	
2.	Розроблення та затвердження плану дипломного проекту.	21.10.19– 22.10.19	
3.	Провести консультації з науковим керівником щодо створення першого розділу.	23.10.19 – 27.10.19	
4.	Розробка розділу 1: Аналіз проблеми просторового мислення	30.10.19 – 11.11.19	
5.	Розробка розділу 2: Методи розвитку просторового мислення	12.11.19 – 22.11.19	
6.	Розробка розділу 3: Приклади ігор для розвитку просторового мислення	23.11.19 – 02.12.19	
7.	Розробка розділу 4: Вибір типу гри для реалізації	03.12.19 – 11.12.19	
8.	Розробка розділу 5: Розробка відеогри	12.12.19 – 22.12.19	
9.	Висновки та оформлення пояснювальної записки дипломного проекту.	25.12.19 – 29.12.19	
10.	Підписання необхідних документів у встановленому порядку.	15.01.20-19.01.20	
11.	Підготовка до захисту та попередній захист дипломного проекту на випусковій кафедрі дипломного проекту	22.01.20 – 31.01.20	

7. Дата видачі завдання: 13.10.2019р.

Керівник дипломного проекту \_\_\_\_\_ Холявкіна Т.В.  
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання \_\_\_\_\_ Семенов Н.М.  
(підпис випускника) (П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту роботи «Кроссплатформна гра для стимулювання розвитку просторового мислення» викладена на 126 с., містить 56 рис., 11 літературних джерел.

**Ключові слова:** UNITY 3D, ПРОЦЕС РОЗРОБКИ, ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ІНТЕГРОВАНА СЕРЕДА РОЗРОБКИ, КРОССПЛАТФОРМНІСТЬ, ПРОСТОРОВЕ МИСЛЕННЯ, МЕТОДИКИ РОЗВИТКУ УЯВИ.

**Об'єкт дослідження:** відеогра, створена за допомогою інструменту Unity3D.

**Предмет дослідження:** просторове мислення.

**Мета роботи:** стимулювання розвитку просторового мислення людини за допомогою відеогри. Розгляд декількох варіантів реалізації відеогри, їх позитивні та негативні сторони. Розробка відеогри.

**Методи дослідження:** порівняльний аналіз отриманих даних для вияву доцільності використання гри для стимулювання розвитку просторового уявлення людини.

**Отримані результати:** у процесі створення дипломної роботи були отримані певні шаблони, які можна використовувати у майбутньому.

**Результати дипломної роботи планується використовувати** для подальшої розробки проектів з метою розвитку певних якостей людини.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	6
ВСТУП.....	7
РОЗДІЛ 1. Аналіз проблеми просторового мислення .....	9
1.1. Функції просторового мислення .....	10
1.2. Просторове орієнтування.....	13
1.3. Стаття у науковому журналі .....	18
ВИСНОВОК ДО РОЗДІЛУ 1 .....	21
РОЗДІЛ 2. Методики з розвитку просторового мислення .....	22
2.1. Методи розвитку уяви у дітей .....	24
2.2. Зв'язок просторового мислення з іншими здібностями людини .....	33
2.3. Розвиток уяви .....	37
2.4. Приклади завдань для розвитку просторового мислення .....	43
ВИСНОВОК ДО РОЗДІЛУ 2.....	52
РОЗДІЛ 3. Приклад відеоігор для розвитку просторового мислення.....	53
3.1. Thomas Was Alone .....	53
3.2. Monument Valley.....	59
3.3. Hitman GO.....	63
3.4. Harmony .....	66
ВИСНОВОК ДО РОЗДІЛУ 3.....	70
РОЗДІЛ 4. Вибір типу гри для реалізації .....	71
4.1. Фаюмський лабіринт .....	71
4.2. Дослідження схожих відеоігор .....	78
ВИСНОВОК ДО РОЗДІЛУ 4.....	90
РОЗДІЛ 5. Розробка відеогри .....	91
5.1. Необхідне програмне забезпечення .....	91
5.2. Розробка архітектури проекту .....	94
5.3. Створення лабіринту .....	102
ВИСНОВОК ДО РОЗДІЛУ 5.....	111
ВИСНОВКИ.....	112
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	113
ДОДАТОК А.....	114

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

GPS (*Global Positioning System*) - система глобального позиціонування .

Swipe / свайп – швидкий рух пальцем по екрану в певну сторону.

JSON (*JavaScript Object Notation*) - текстовий формат обміну даними, заснований на JavaScript. Як і багато інших текстові формати, JSON легко читається людьми.

SDK (*Software Development Kit* ) - це певний набір засобів для розробки.

JDK (*Java Development Kit* ) - це набір застосунків для розробки програм на мові Java.

IDE (*Integrated Development Kit*) – інтегрована середа розробки

GUI (*Graphical Sser Interface*) - графічний інтерфейс користувача

AR (*Augmented Reality*) - доповнена реальність

VR (*Virtual Reality*) – віртуальна реальність

FBX (*Filmbox*) – технологія і формат файлів розроблені які використовують у 3D моделюванні для забезпечення сумісності різних програм.

AI (*Artificial intelligence*) – розділ комп'ютерної лінгвістики та інформатики, що опікується формалізацією проблем та завдані, які подібні до дій людини. Проще кажучи штучний інтелект.

## ВСТУП

У сучасному світі неможливо уявити людину яка не здатна до базових дій зі своєю уявою. Вона використовується не лише для творчих професій таких як художників, дизайнерів, візажистів, кравців, перукарів, фотографів, ювелірів, мультиплікаторів, ілюстраторів, композиторів, флористів. А також для більш технічних професій як інженерів, архітекторів, програмістів.

Але як на мене, на сьогодні для кожної людини є один життєво необхідний навик який повністю пов'язаний з уявою. Це просторове уявлення, може здатись що воно потрібно хіба що для занять по геометрії чи 3д моделюванню. Зараз, міста розрослись до неймовірно великих, за своїм масштабом, переплетіння споруд. Вони настільки великі що наврядчи знайдеться людина яка буде досконально знати усе своє місто. Більшість людей взагалі були тільки в окремих районах, і дуже поверхнево уявляють собі карту рідного краю.

Хоча люди і намагаються будувати їх притримуючись певної системи, розбиття на вулиці, райони і тд. Але навіть так, це не дуже допомагає орієнтуватися у невідомій частині міста, а деякі навіть на своїй рідній вулиці можуть трохи заблукати.

Звичайно зараз існують безліч паперових и електронних карт. На останніх при увімкненому GPS навіть буде показувати у яку сторону дивиться людина відносно карти. Проте навіть так у великої кількості людей виникають проблеми з орієнтуванням у місті. На мою думку це пов'язано з просторовим уявленням людини.

Тобто, людина у якої є проблеми з уявою, не може нормально та швидко скласти у своїй голові карту місцевості то співвідносити своє місцеположення на ній. Не може побудувати шлях свого руху, запам'ятати певні орієнтири і так далі. На мою думку це дійсно є проблемою для сучасної людини, яка майже кожен день виходить у місто з метою доїхати

до роботи, магазину, кінотеатру і тд.

Є велика кількість способів та методик для розвитку просторового уявлення. Більшість методів спрямовані на розвиток у дошкільному та шкільному віку. Адже саме в цей час людина вчиться найкраще и це буде ідеальним моментом для створення необхідної бази вмінь для кожної людини. Саме на міцному фундаменті потім будуть будуватись нові вміння та навички, отже це ідеальний момент для розвитку просторового уявлення.

Наприклад для дітей віком до 3 років, можна запропонувати простенькі задачки для їх розвитку:

- Малювати різні фігури створені із точок, самостійно або повторюючи вже готовий приклад.
- Знаходити одинокі точки та з'єднувати їх у як найбільшу кількість прямих ліній.
- Будувати із паличок різні фігури, букви, будівлі та інші речі.

Для більш чіткого уявлення того що саме потрібно реалізувати у даній грі, буде проведено порівняльний аналіз вже існуючих ігор для різних платформ. Виявлення їх недоліків та переваг.

Для реалізації відеогри планується використовувати або спеціальні бібліотеки для певних мов програмування, або використовувати вже готовий ігровий двигун. У кожного з методів є свої плюси і мінуси, тому перед тим як вибрати шлях створення гри, необхідно чітко проаналізувати усі моменти які необхідні саме для нас, і провести порівняння обраних методів.



## РОЗДІЛ 1. Аналіз проблеми просторового мислення

Перш ніж почати розробляти гру, необхідно виконати декілька кроків, які допоможуть у майбутньому зекономити час, правильно виставити пріоритети, досягти поставленої мети, а саме:

- Визначення з причиною проблем людини з розвитком просторового уявлення.
- Проаналізувати існуючі методи.
- Провести розбір конкурентів, а саме відеоігор зі схожою тематикою.
- Визначитись зі шляхом реалізації гри

Перш за все необхідно визначити що саме розуміється під поняттям «Просторове уявлення» - це уявлення, у котрих мають місце бути просторові відносини між різними предметами. Їх формою, розміром, взаємним розташуванням, рухом.

Рівень типовості й схематизації просторового образу підпорядкований не лише формою самого предмету а й від завдання діяльності, що буде реалізовано людиною, і в якій використовуються загально прийняті засоби просторового аналізу. Це можуть бути карти, малюнки, схеми, та інші види зображень.

Сам по собі термін «уявлення» на відміну від уявлення в загальному його значенні, був уведений для опису складної інтелектуальної діяльності зі складання необхідних образів та оперування ними. Далі його стали використовувати для позначення процесу випадкового використання образу й теоретичного оперування ним у процесі вирішення різноманітних завдань.

Кафедра КІТ				НАУ 20 26 27 000 ПЗ			
Виконав	Семенов Н.М.			Аналіз проблеми просторового мислення	Літера	аркуш	аркушів
Керівник	Холявкіна Т.В.					9	12
Консульт.					УС 211М 122		
Н. контроль	Райчев І.Е.						

## 1.1. Функції просторового мислення

Одна і напевно найважливіша з функцій просторового уявлення, складається в тому зоб забезпечити просторові властивості і відносини, вміти проводити з ними певні дії та операції при вирішенні різноманітних завдань. Ці завдання частіше за все залежать від орієнтації об'єкту в фізичному та теоретичному просторах. Просторові відносини означають відносини між об'єктами простору або між просторовими ознаками цих об'єктів [8].

Вони виражаються в напрямку:

- вперед і назад
- вгору і вниз
- вправо і вліво

Відстані:

- далеко і поблизу

Їх відносинах:

- ближче і далі,

Розташуванні:

- всередині

Довжині об'єктів в просторі:

- високий-низький
- довгі-короткі

Основними якісними показниками просторового мислення є:

1. тип операції над просторовими образами та широта операції з урахуванням використовуваної графічної основи.
2. Повнота зображення, переважне відображення в ньому форми, розміру, та просторового положення.
3. Стабільна система відліку, просторова орієнтація з довільної точки

відліку.

Також, важливою ознакою розвитку просторового уявлення є широта функціонування та повнота образу. За широту функціонування (оперування) частіше за все розуміють ступінь свободи маніпулювання просторовими зображеннями в процесі використання різних графічних матеріалів. Повнота зображення це відповідність його реальному об'єкту. Вона характеризує сукупність елементів зображення, їх динамічність та зв'язок.

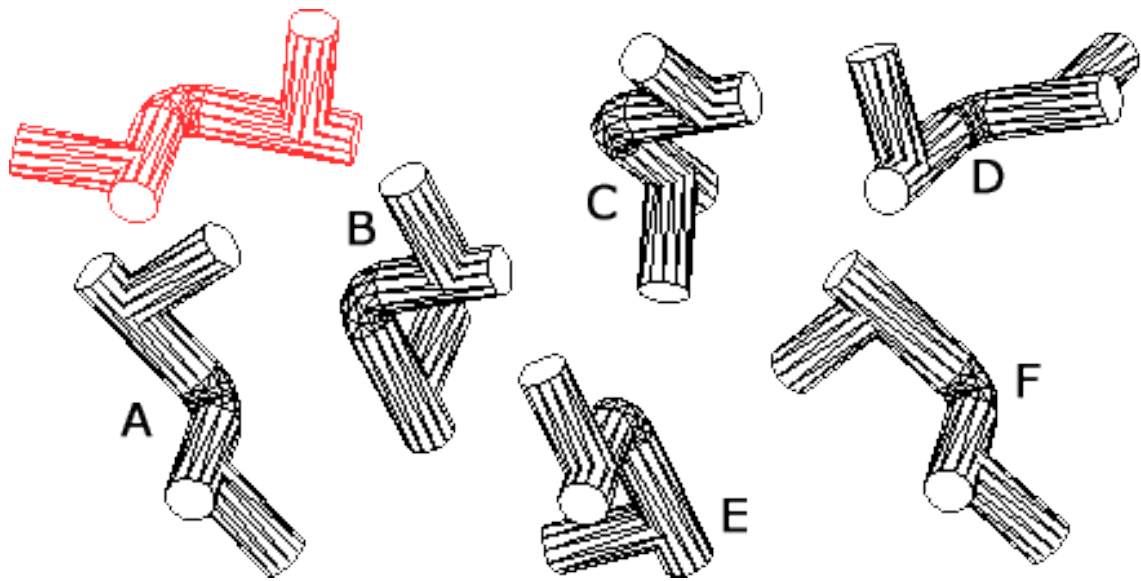


Рис. 1.1. Приклад просторового оперування

На рис 1.1. можна побачити що мається на увазі під оперуванням об'єктом у 3д просторі. Та його проєкції у 2д вимір. Кожне зображення має однакові розміри відносно одне одного, але в залежності від куту нахилу та спостереження, їх видимі образу можуть дуже різнитись. Задача просторового уявлення у впевненому оперуванні такими об'єктами ти повне розуміння їх проєкцією під будь-яким кутом спостереження.

Зміст і характер просторового уявлення, його функції визначаються умовами, в яких воно формується, проявляється і вдосконалюється. Основна оперативна величина просторового уявлення це зображення яке представляє просторові характеристики об'єму:

- Форма
- Розмір
- Розташування їх на площині
- Розташування у просторі відносно заданої точки відліку

Саме ці характеристики відрізняють просторове мислення від інших форм образного уявлення. Де виокремлення просторових характеристик не є головною дією.

Аналіз просторового мислення дав змогу описати його структуру, розділити це складне утворення на складові елементи. Виявити співвідношення різних рівнів розвитку окремих елементів і системи в прямому їй розумінні. Визначити їх відносну ієрархію. Створення просторових образів здійснюється на графічній основі шляхом їх ментального перероблення. Операції над просторовим зображенням здійснюється в умовах графічних завдань, іноді з відхиленням від вихідної наочної основи. Перетворення мислених образів частіше за все відбувається одномоментно в деяких напрямках, чи навпаки тільки в одну сторону в один проміжок часу. Це на мій погляд визначає складність перетворень, котрі виконується в уяві людини, а отже й основна структура просторового уявлення.

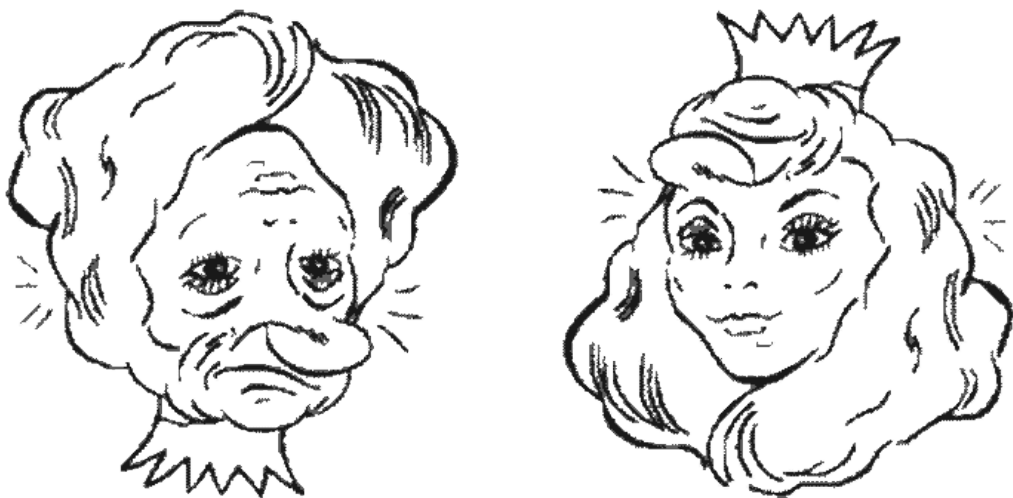


Рис. 1.2. Приклад малюнку для тренування

На рис 1.2. приведений гарний приклад малюнку який при різних кутах погляду виглядає зовсім по іншому. З одного боку старій чоловік з великим носом. А з іншої молода дівчина з короною та пишним волоссям. Навіть людина з досить непоганим просторовим мисленням навряд чи зможе з першого погляду помітити це якщо залишити лише ліву частину зображення. Потрібно дійсно гарне уявлення щоб не повертаючи голови зуміти побачити другий варіант.

Отже ми розглянули один з основних видів просторового уявлення, в якому головним моментом є оперування певними характеристиками об'єкту, самому по собі або відносно інших. Дізнались що таке уявлення, та як від інших його проявів відрізняються просторове мислення. Визначили що зміст і характер просторового мислення залежить від умов в яких формується об'єкт.

## **1.2. Просторове орієнтування**

Далі розглянемо другу складову частину просторового мислення, як просторове орієнтування. Ці два поняття невід'ємні одне від одного. Адже неможливо нормально орієнтуватися в нашому 3д просторі, якщо у людини немає базових навичок просторового мислення. Відносно розташування об'єктів один від одного, положення людини серед них, напряму погляду та руху, кінцева точка маршруту яка постійно змінює своє місцеположення відносно руху людини и тд.

Добратись від пункту А до пункту Б для деяких людей є досить непростим завданням (рис 1.3.). Цю проблему жартівливо називають «Топографічним кретинізмом», хоча насправді нічого смішного в цьому немає. В чому може критись корінь проблеми дезорієнтації в просторі і як з цим боротись? Далі спробуємо відповісти на ці питання.

В яких ситуаціях часто проявляється така проблема? Наприклад:

- 1) Людина паркує свою машину на стоянці перед великим магазином

і потім не може знайти її, блукаючи з візком серед машин.

- 2) Ще важче знайти залишене авто в місті, особливо на невідомій вулиці. Адже в одному районі може бути багато схожих проспектів, площ, а десь ти повертав праворуч, або ліворуч и все в голові переплутується.
- 3) Гуляючи невідомими стежками міста, наче розумієш куди йдеш, робиш лише декілька поворотів, а потім не можеш повернутися на початок, адже такий легкий маршрут, в зворотному напрямку виглядає зовсім не так як ті його собі уявляв.

Й таких неприємних моментів та ситуацій можна привести велику кількість. Що лише підкреслює початкову думку, про існуючу проблему просторового орієнтування а отже и просторового мислення у людини.

Як пояснив гештальт-терапевт Серж Гінгер – почуття орієнтації в просторі не пов'язано з якимось одним органом сприйняття, на відміну від інших 5 почуттів. Воно являється результатом збору інформації декількома органами, а потім їх аналіз зроблений мозком. Тобто це не закладено в людину з народження, а отримується на базі життєвого опиту то формується як необхідний інструмент для виживання людини в природі або межах міста (села).

Як повідомляє той же лікар, чоловічий гормон тестостерон сприяє більш активному розвитку правої напівкулі мозку людини. Котра по багаторазовим досліддам та перевіркам відповідає за просторове уявлення людини. Таким чином у чоловіків в цілому біль розвинуте внутрішнє відчуття просторову, та орієнтація в ньому. Але на перевіс цьому у жінок більш розвинута зорова пам'ять. Також це підтверджується досліддами, у яких жінки при підвищенні тестостерону (наприклад під час місячин) починають краще орієнтуватися на місцевості по карті, або без неї.

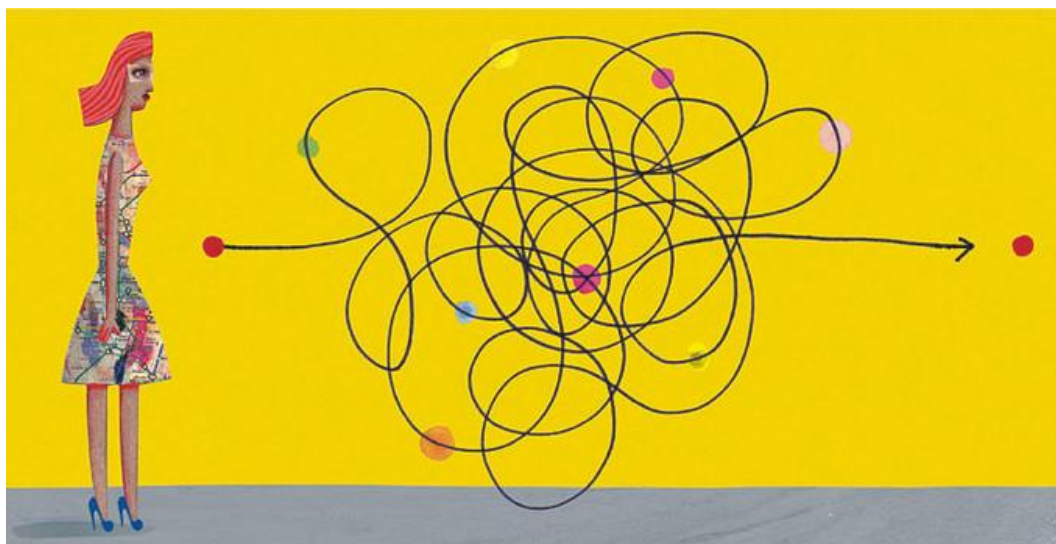


Рис. 1.3. Важкий пошук шляху

Чи впливає на цю характеристику людини, еволюція? Орієнтування на невідомій місцевості це не теж саме що й вміти орієнтуватися по карті. Все як раз навпаки, це відчуття допомагає нам у тих випадках, коли карти або GPS немає під рукою. Наприклад, це відчуття спроможне підказувати людині де в невідомому місті, новій вулиці і тд, знайти ресторан, щоб пообідати, туалет або готель. У жінок та мужчин різний підхід для вирішення цих завдань. Чоловіки керуються відчуттям напрямку, а жінки зазвичай завдяки конкретним орієнтирам. Це зумовлено різною роллю полів у давні часи. Жінки рідко відходили далеко від стоянки, або селища, особливо в одиночку під час збору трав, ягід і тд у лісі. А чоловіки в більшості своєму були мисливцями і дуже часто виникала ситуація коли під час переслідування здобичі немає часу помічати якісь орієнтири чи гарно запам'ятовувати шлях. Але при цьому намагаються як найшвидше дістатись до свого дому.

Тобто деякі моменти пов'язані с простором мисленням та орієнтуванням на місцевості дійсно пов'язані з нашим минулим, тим укладом життя який існував багато тисяч років. Ці звички та інстинкти залишились в нас ще з тих часів. Звичайно це не означає що усі жінки

погано орієнтуються и не мають почуття напрямку, а чоловіки мають погано зорову пам'ять та не бачать орієнтирів. Мається на увазі що кожен з нас більш схильний до чогось. Але кожен може розвинути в собі ті якості, над якими він буде важко працювати, і з кожним роком ця різниця стає все меншою и меншою[10].

Здібність до орієнтування можна розвивати, але е важко для тих хто схилений до паніки й применшує свої можливості через якісь психологічні проблеми, блокери, стереотипи чи дитячі травми. Довіра до себе дуже крихка річ, яку дуже легко подавити в дини з дитинства, наприклад:

- Дуже авторитарними батьками, які просто не дають дитині досить простору для прийняття власних рішень.
- Віковими стереотипами де жінкам внушалось те, що чоловіки перевершують їх у будь яких справах, і ці патріархальні стереотипи й досі сильно впливають на сучасних людей у всіх країнах світу.
- Для багатьох із нас, така «мимовільна розгубленість» - спосіб утримання в дитинстві, щоб відчутти себе безпорадним і ухилитися, хоча б не на довго, від будь якої відповідальності.

Доки це виникає не дуже часто і більше відчувається як гра, в цьому немає нічого поганого. Але чим більше і довше це трапляється тим більше ми до цього звикаємо і це може звужувати наші горизонти, наше сприйняття реальності з його проблемами та відповідальністю перед собою та іншими. Тоді приходить час зрозуміти, що ми дорослі, самостійні люди й в наших можливостях знаходити рішення для будь-яких завдані які ставить перед нами життя. И для цього необхідно навчитись відноситись до себе с довірою та повагою, так як все починається з внутрішнього світу людини.

Що робити для того щоб розвити цю необхідну здібність в собі?



Якщо дуже коротко то потрібно зберігати спокій в подібних ситуаціях, адже трошки подумавши можна прийти до виводу що заблукати в місті або загубити машину майже неможливо. З орієнтуванням у місті завжди можуть допомогти прохожі, яких на будь який вулиці міста завжди можна зустріти. Чим спокійніше відносишся до проблеми тим легше з нею справитись. Наприклад проблему з орієнтуванням можна розглядати і з оптимістичної позиції, це як гарний привід зайвий раз прогулятись і більш детально розглянути те що нас оточую. Вулички рідного (чи іншого) міста завжди можуть містити якісь цікаві дрібниці які при швидкому, мимохідному, проходженні не помітні, і неможливо скласти повну картину.

Забудьте про GPS навігатор. Цей маленькій пристрій або програмне забезпечення на вашому смартфоні яке підказує, коли потрібно повернути праворуч, а коли ліворуч, може бути вірним другом при будь-яких подорожах. Однак має сенс використовувати його лише у важких маршрутах. Інакше наш внутрішній компас ризикує деградувати від такого «постійного» використання. Адже він як і будь-які наші здібності потребує постійних тренувань. Більше грайте і різні ігри які пов'язані з орієнтуванням на місцевості. Наприклад в слідопитів, чи пошук кладу по карті, або навіть піти на курси спортивного орієнтування (більше підходить для дорослих аніж попередні ідеї) – усе це гарні способи для внутрішнього компасу, бо в ігровій формі все вчиться набагато легше і приємніше.

Міжнародна група вчених на чолі з Барбарою Ландоу з Університету Джонса Хопкінса виявили що вміння орієнтуватися має також генну природу. Відповідна стаття була опублікована в *Proceedings of the National Academy of Sciences*. Це один з найпопулярніших американських журналів для публікації оригінальних наукових досліджень в різних областях. Головним чином в біології та медицині, а також фізики та соціальних наук. У 2009 році увійшов у список «100 найвпливовіших журналів біології і медицини за останні 100 років», і зайняв 3-є місце в категорії

### 1.3. Стаття у науковому журналі

«Молекулярна і клітинна біологія, біотехнологія та міждисциплінарні журнали» Володіючи великим авторитетом в науковому світі журнал PNAS має велику відвідуваність своєї онлайн версії в інтернеті, майже 11.6 млн хітів в місяць. Інакше кажучи туди публікують дійсно перевірені дослідження які проходять декілька перевірок адже вони адже переймаються за свою репутацію.

Отже в даній статті Ландау каже «Наскільки нам відомо, наша робота являється першим свідченням зв'язку генів з відсутністю системи що дозволяє орієнтуватися в просторі». Даний вивід вчені зробили, досліджуючи людей з рідкими генетичними хворобами, наприклад хворобою Вільсона (синдром «лиця ельфа»). Вона розвивається в результаті хромосомної патології, кола мається порушення 7-мої хромосоми, що призводить до клінічних випадків гіпокапнії (збиткова кількість вуглекислоти в крові, по суті, хронічному отруєнню). Це в свою чергу приводить до загальної затримки розумового розвитку людини, хоча деякі області її інтелекту при цьому непогано розвинені.

Вчені провели ряд досліджень, метою яких було виявити наскільки гарно люди які страдають цією хворобою можуть орієнтуватись в просторі. В ході експерименту випробувані мали знайти певний предмет, який був схований під темну тканину у них на очах. Задача ускладнюється тим, що після того як предмет був схований, їм зав'язували очі і в продовж 10ти секунд крутили їх, щоб дезорієнтувати їх відносно приміщення. Крім того в кімнаті практично не було ніяких орієнтирів (в одному експерименті стіни мали чорний колір в другому одна з стін була синього кольору). Але при цьому кімната мала прямокутну форму, що повинно було полегшити пошуки.

Автори цієї статті порівнюють з дитячою грою «Прищепи хвіст віслюку» (на англ. Pin the tail on the donkey), яка популярна в англійських

країнах. В ній гравець має з зав'язаними очима правильно приколоти хвіст до зображення віслюка.



Рис. 1.4. Гра «Прищепи хвіст віслюку»

Також її назву вживають висловлюючись про марну діяльність або про якого небудь інваліда.

Людина, яка має нормальне просторове уявлення а отже и простору орієнтацію, частіше за все без усяких проблем справиться з цим завданням, просто закріпившись за формою кімнати, і пам'ятаючи про те, де був схований предмет. А ось людина яка хворіє хворобою Вільсона проявляю повне невміння орієнтуватися в цій ситуації.

Контрольна група із здорових людей підтвердила що знайти предмет

було досить легко. На основі експерименту вчені можуть зробити вивід, що причина порушення нормальною здібності до просторового орієнтування може також бути пов'язана с генетичною патологією.

Як сказали вчені «Хоча ми досить далекі від розуміння зв'язку між конкретними генами, які відсутні у людей, вражених хворобою Вільсона, і особливостями хвороби, наприклад неспроможністю нормально орієнтуватися в просторі, ясно, що саме недостача генів має вплив на мозок людини» - стверджує Барбара Ландау.

Отже враховуючи усі дослідження та попередні тези можна прийти до висновку що просторове мислення, а також похідна від нього – просторове уявлення, це здібність людини яку вона може і повинна розвивати. Без цієї здібності людина не може впевнено почувати себе на вулиці, боячись загубитись у власному місті. Саме стимулюванню розвитку просторового мислення і, зокрема, просторового орієнтування буде присвячений цей диплом.

## ВИСНОВОК ДО РОЗДІЛУ 1

В першому розділі було розглянуто проблему орієнтування у просторі людини у наш час. Було наведено багато прикладів підтверджуючих що проблема має місце бути.

Отже, ми можемо зробити висновок, що у сучасному суспільстві дійсно є проблема с просторовим мисленням. Звичайно на це впливають різні фактори, які іноді не залежать від людини. Але у більшості випадків ця проблема пов'язана з недостатньою кількістю вправ для розвитку просторового мислення.

Забудьте про GPS навігатор. Цей маленькій пристрій або програмне забезпечення на вашому смартфоні яке підказує, коли потрібно повернути праворуч, а коли ліворуч, може бути вірним другом при будь-яких подорожах. Однак має сенс використовувати його лише у важких маршрутах. Інакше наш внутрішній компас ризикує деградувати від такого «постійного» використання. Адже він як і будь-які наші здібності потребує постійних тренувань.

## РОЗДІЛ 2. Методики з розвитку просторового мислення

Якщо коротко пройтись по усім пунктам то отримуємо наступне. Має сенс взятись за розвиток просторового мислення ще с дитячих років людини. Адже саме в цей період дитини має найкращі здібності для інтуїтивного розвитку цієї здібності. Можливість орієнтування у просторі забезпечується взаємодією:

- Зорового
- Слухового
- Рухово-кінестетичного

аналізаторів при різних видах діяльності людини. Зазвичай у цьому віці вони спрямовані на активне пізнання навколишнього світу.

При розвитку просторового мислення, дитина, також формує відчуття будови свого тіла, а з більшою практикою, дітей у використанні цих якостей та вмій, зі зміною порядку об'єктно-ігрової дії, пов'язаною з подальшим вдосконаленням вмій руху, та контролю тіла. Створені просторові ідеї відображаються та надалі розвиваються в об'єктно ігровій, побутовій, образотворчій і конструктивній діяльності дитини.

Але все це кількісна складова розвитку, ця стадія лише дозволяє відточити ті навички що є, ти бути біль впевненим у них. Якісна зміна у формуванні просторового мислення буде пов'язана з розвитком мови. З розумінням і постійним вживанням дітьми словесних значень просторових відносин, виражених приводами, прислівниками.

Оволодіння знаннями про простір потребує навичок розрізняти та виділяти просторові відносини та ознаки. Вміння правильними словами їх позначити, орієнтуватися у просторових відносинах і під час різних трудових дій, які залежать від просторового мислення.

Кафедра КІТ				НАУ 20 26 27 000 ПЗ			
Виконав	Семенов Н.М.			Методики з розвитку просторового мислення	Літера	аркуш	аркуші
Керівник	Холявкіна Т.В.					22	30
Консульт.					УС 211М 122		
Н. контроль	Райчев І.Е.						

Велику роль у розвитку просторового мислення дитини грає ліплення та конструювання (наприклад різних розвиваючих конструкторів та задач на просторове сприйняття, або ліплення з пластиліну глини і тд), включення правильних дій дітей і словесних позначень в мову.

На другому етапі розвивають здатність виконувати різні рухові дії тілом, відповідно до мінливих умов виконання. Наприклад кидання м'яча у мішені, які рухаються в певних напрямках з різною швидкістю.

На останньому етапі розвивають здатність дитини використовувати саме ті дії, які найбільш відповідають різним, раптово виникаючим, ситуаціям. Для розвитку такої здатності вдаються до різних рухомих і спортивних ігор. У школах часто практикують гру під назвою «Вибивний» (рис. 2.1.) де між двома головними гравцями, які будуть кидати м'яч, на середній дистанції стоїть купа інших гравців. Ціль гри для двох головних – попасти м'ячем по усім людям які стоять між ними. А задача інших від цих м'ячів ухилятися. Ця гра як раз викликає різні раптові ситуації у яких дітям необхідно діяти швидко та впевнено, використовуючи усі можливості свого тіла якими вони змогли оволодіти за цей час. А також мотивує для подальшого розвитку цих необхідних у житті та грі, навичок і вмінь.



Рис. 2.1. Гра «Вибивний»

Якщо більш конкретно розглядати кожен етап росту дитини, та розвиток її уявлення то є сенс розглянути кожен рік більш конкретно. Адже для кожного віку існує свій рівень розвитку просторового уявлення. Там де колись були будівлі з дерев'яних кубиків, по мірі дорослішання може вийти детальна тривимірна модель хмарочосу із деталей конструктору.

## **2.1. Методи розвитку уяви у дітей**

Дітки до 3-х років. З народження й до 2-3 років дитина вчаться обробляти сигнали, які поступають до її головного мозку від органів відчуття – дотику, слуху, зору, нюху, відчуття температури. І об'ємне сприйняття засновано якраз на зорових, слухових та рухомих аналізаторах. Десь приблизно у 3 роки в дитини формується розуміння свого положення в просторі і свідомий навик оцінювання й вияву взаємного положення різних об'єктів. Малюк вже має здібності для вирішування простих завдань, змінюючи місцеположення предметів, але не торкаючись їх структури (задачі на переміщення) [11].

Розвиток просторового уявлення у дитини в три роки, неможливо відокремити від розвитку мислення в цілому. Адже це комплексний розвиток де одне допомагає іншому і навпаки. Тому не раджу нехтувати цим й вчити дитину думати в цілому а не лише в окремих аспектах. При цьому завдання мають бути досить простими й по можливості у формі гри, веселого завдання або посильного випробування.

Дитина до 3-х років вже може:

- відрізнити горизонтальні, вертикальні, нахилені лінії
- відрізнити криві і прямі лінії
- бачити основні геометричні фігури
- знаходити точки перетину різних ліній

Для розвитку просторового мислення дитини віком до трьох років рекомендується:



- знаходити точки та об'єднувати їх
- будувати із паличок різні букви, фігури та будівлі
- малювати фігури з точок повторюючи за оригіналом чи випадково
- вчити дитину малювати
- Аплікація, ліпка з пластиліну

### *Молодші і середні дошкільнята (3-5 років)*

У цьому віці дитини здібна вирішувати просторові задачі шляхом різних трансформацій: накладання, поєднання, перегрупування складових частин, додавання або видалення елементів. У дитини формується інтуїтивне розуміння відносин між предметами: «Однакові», «різні», «більший», «менший», «справа», «зліва», «між». Усвідомлене оперування цими термінами приходить пізніше - в старшому дошкільному і молодшому шкільному віці.

Дитині 3-5 років вже посилам:

- повторювати прості узори й орнаменти із ліній
- порівнювати довжину відрізків, розміри фігур
- поєднання точки по цифрам, щоб отримати малюнок, звичайно якщо дитина вже знаю порядок рахунку
- відрізнити прямий, гострий і тупий кути

Як стимулювати розвиток просторового мислення в 3-5 років?

- Ігри на формування тривимірних представлень – «Вмісти кішку» де необхідно вибрати тваринку и уявити, в який предмет вона може вміститись
- «Розстав правильно» (необхідно розташувати предмети за вказівкою дорослого), «Склади маршрут»
- завдання на маніпуляції з фігурами (як буде виглядати квадрат за

колом, піраміда на кубі)

- картинки, що вимагають охарактеризувати місце розташування одних об'єктів відносно інших ( «де знаходиться кішка? - за будинком»)
- Вирізуйте з паперу симетричні предмети і сніжинки, домальовуйте другі половинки картинок

Також займайтесь орігамі (рис.2.2), складайте пазли і геометричні аплікації.

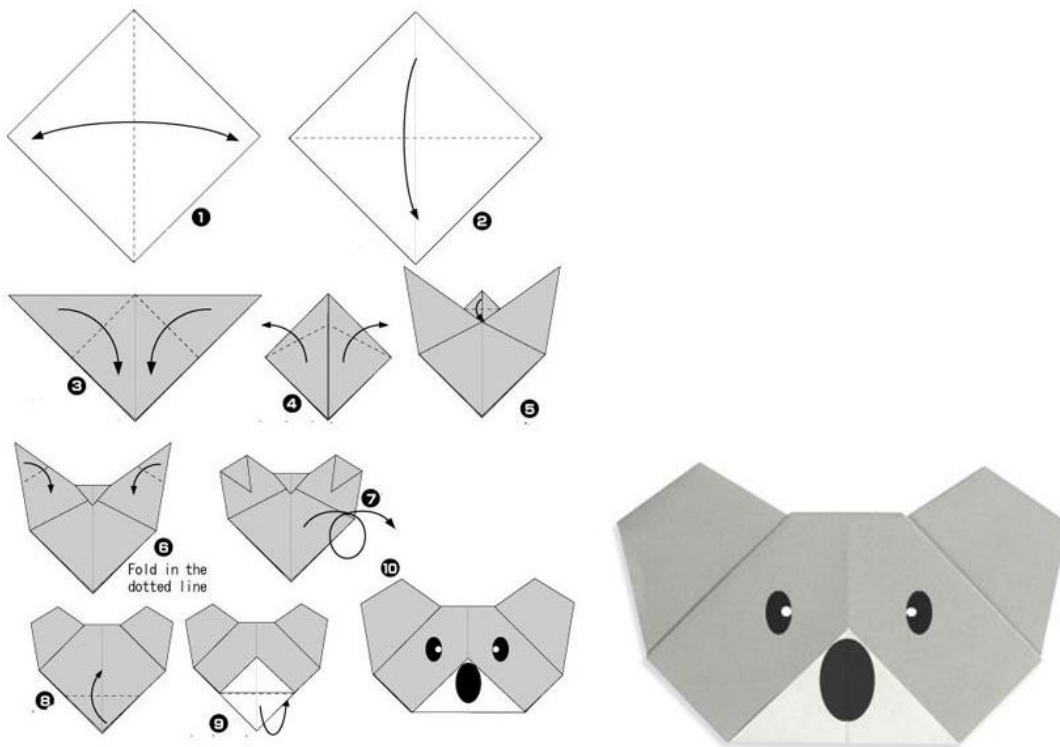


Рис. 2.2. Орігамі

### *Дошкільнята (5-7 років)*

У цьому віці дитина вже може вирішувати досить складні просторові завдання, наприклад такі, в яких початкове положення, зовнішній вигляд, структура об'єкта неодноразово змінюється. Виконуючи такі завдання, дитина здатна не тільки розглядати і пересувати тривимірні об'єкти, але і перебудовувати, перетворювати і «покрещувати».

У 5-7 років діти повинні вміти:

- розрізняти поняття розміру і форми
- вільно оперувати просторовими відносинами між предметами (великий-маленький, ближче-далі, вгору-вниз, над-під, за-перед і т.д.)
- розрізняти ліву і праву руку, сторону
- розуміти заданий напрямок, щоб переміщатися відповідно до нього
- подумки співвідносити між собою об'єкти, що знаходяться в полі зору
- орієнтуватися в двомірному просторі (площина з довжиною і шириною)

Названі освітні вимоги пред'являють до майбутніх учнів школи. Так, вони не обов'язкові, адже в перший клас дитини візьмуть в будь-якому випадку. Але для полегшення адаптації до школи і розуміння навчального матеріалу перераховані вміння дуже знадобляться першокласнику.

Як розвивати просторове мислення в 5-7 років?

З цим завданням допоможуть впоратися:

- гри на уважність. Наприклад, дитина пильно оглядається в приміщенні, де він знаходиться, і запам'ятовує навколишні його предмети. Дорослий ставить запитання з використанням слів, які вказують на місце розташування, наприклад, «якого кольору стіл зліва від тебе?»
- графічні диктанти
- цікаві завдання на об'ємне мислення
- спеціальні розвиваючі ігри
- повсякденні побутові завдання

Пропонуйте дитині настільні ігри, такі як Танграм (рис. 2.3.), «Геометричний», шашки, шахи, нарди. Разом намалюйте плани приміщень

і місцевості, грайте в пошук «скарбу» за допомогою карти-плану.

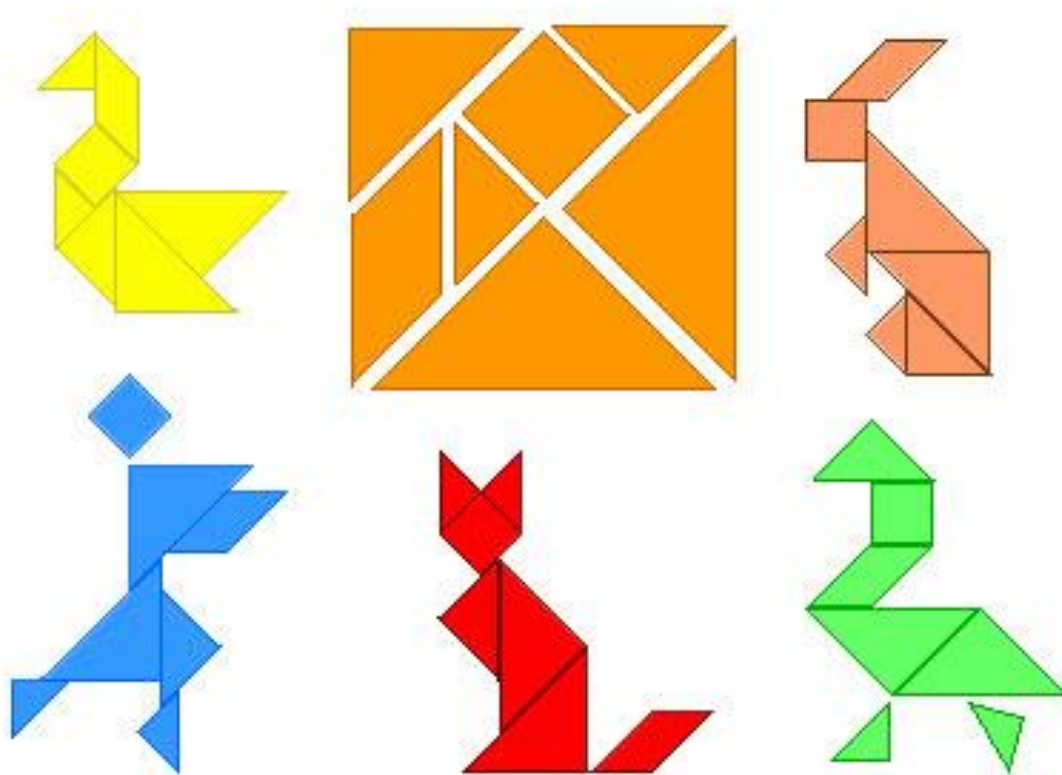


Рис. 2.3. Гра «Танграм»

### ***Молодші школярі (7 + років)***

Розвитком тривимірного мислення школярів займаються вчителі - на уроках математики, навколишнього світу, праці, малювання. У молодших класах діти, в основному, вчать мислити «на площині», але в міру дорослішання завдання стають «тривимірними», - і бажано, щоб до такого переходу ваш школяр був готовий.

### **Дитина старше 7 років вмiє:**

- моделювати стан об'єктів в просторі - представляти, не вдаючись до фізичного переміщення предметів;
- звертати увагу на просторові відносини між предметами, їх взаємне розташування і його зміна;

- бачити картину з різних точок зору, визначати чи вибирати правильну проекцію тривимірного предмета - подивитися на фігуру з одного боку і зрозуміти, як вона виглядає з іншого. До 9 років школяр вже здатний на більше складні операції, пов'язані з координуванням перспектив, наприклад, визначити виходячи з різних проекцій, як виглядає тривимірна фігура;
- формулювати в усному мовленні просторові характеристики об'єкта.

### *Як тренувати просторове мислення в 7 років і старше?*

- завдання на співвіднесення предметів в просторі і на площині відносно один одного. Наприклад, «намалюй стакан, тарілку і ніж так, щоб ніж виявився праворуч від тарілки і зліва від склянки»;
- завдання на перекреслення фігур: просте копіювання точь-в-точь, перемальована із зразку, зміною розміру або точним масштабуванням, відтворення зображення по пам'яті;
- малювати маршрути - наприклад, свій шлях від будинку до школи або маршрут казкового героя, що рухається по уявному замку;
- творчі завдання з будь-якими видами конструкторів - не по схемі, а за допомогою власних навичок; при цьому здайте вихідні параметри, наприклад, середньовічний замок з трьома вежами, двоповерховий гараж з паркуванням і т.д.

Грайте з кубиками «по-дорослому» - створюйте складні конструкції і робіть з ними метаморфози, складайте фігури по «тіні» (схематично зображення однієї, двох або трьох проекцій). Для цих цілей знадобиться розвиваючий ігровий набір «Геометрія в кубиках», або блоки Дьенеша / «цеглинки» (рис 2.4.) плюс власна фантазія. Складайте разом з дитиною

тривимірні пазли, будуйте об'ємні моделі предметів.

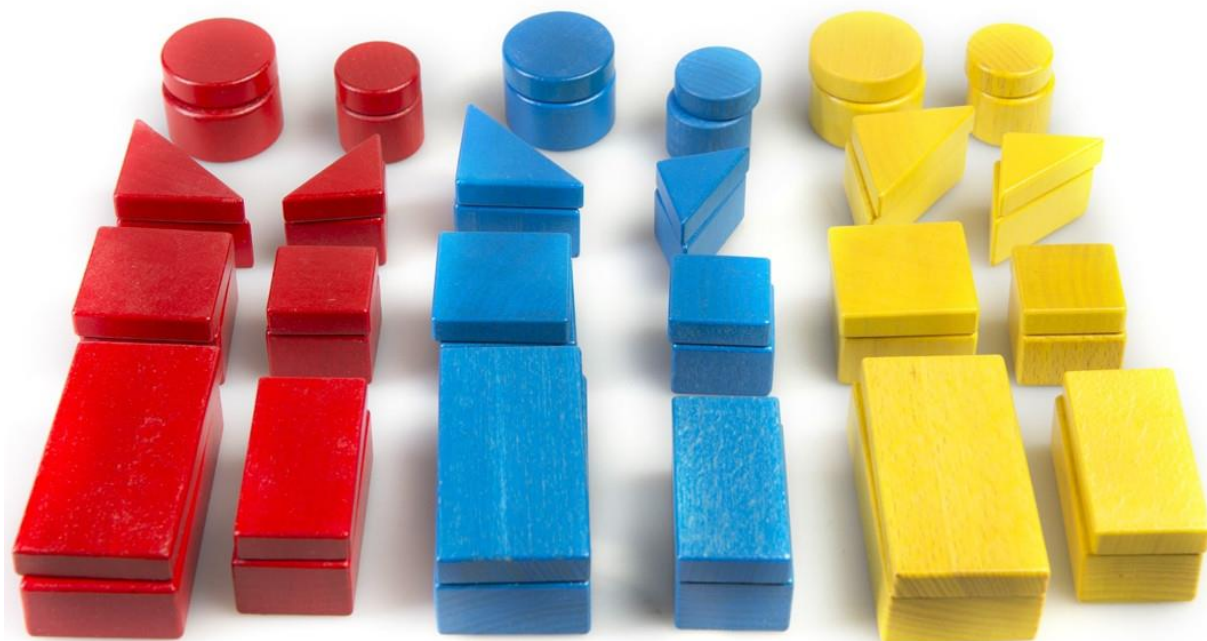


Рис. 2.4. Гра «Дьенеша»

***Об'ємне, тривимірне, образне мислення, просторову уяву***

Я постарався трохи розібрати відмінності в поняттях. З теорією все зрозуміло, тому можна перейти до практичних рекомендацій.

Просторове (об'ємне, тривимірне) мислення - це процес, при якому людина усвідомлює просторові характеристики об'єкта (розмір, форму, взаємозв'язок компонентів і місце розташування) і подумки вчиняє дії з цим об'єктом (наприклад, передаючи або перетворюючи його).

Об'ємне мислення використовує інформацію, які надходить з органів почуттів, обробляє їх, активно використовуючи пам'ять і логіку. Воно безпосередньо пов'язане з образним мисленням, адже обидва різновиди користуються візуалізацією - навиком «намалювати в розумі» картинку. Висувати просторові гіпотези нам допомагає уява.

## **Що розвиває - тривимірне мислення або уяву?**

Відмінність в деталях. «Картинка» тривимірного мислення - гранично достовірна. «Продукт» образного мислення частіше уявний, спотворений, що сприймається крізь призму власного досвіду або емоційно-забарвлений (наприклад, якщо об'єкт подобається - він видається в уяві більш привабливим).

Просторове мислення тісно пов'язане з уявою: ми задіємо саме його, коли подумки міняємо, повертаємо, пересуваємо в просторі тривимірні моделі об'єктів. Ми не бачимо цих перетворень - але фантазуємо про них. Розвиваючи тривимірне мислення у дитини, батьки тим самим стимулюють, дають поштовх уяві, в тому числі творчому.

Людина від природи наділена тривимірним мисленням, але у одних з нас воно розвинене краще, ніж у інших. На думку вчених, частково це пов'язано зі спадковістю, але іноді - з відсутністю відповідного досвіду і «тренувань». Починати займатися можна буквально з 3-5 років, але необхідно орієнтуватися на вікові етапи розвитку просторового мислення .

### **Як просторове мислення допоможе майбутньому першокласнику**

Уміння задіяти об'ємне мислення, в побутовому розумінні - «орієнтуватися в просторі», - навик гранично практичний.

### ***Дитина з розвиненим просторовим мисленням:***

*Добре орієнтується на місцевості.* Він швидко запам'ятає розташування кабінетів у школі, не заблукавши в коридорах, самостійно знайде вихід з торгового центру або музею, буде впевнено почувати себе на екскурсіях з класом і при відвідинах театру, вийде на потрібній зупинці з транспорту (а якщо помилиться - легко зрозуміє, як її виправити, і зрозуміло пояснить дорослому, де він знаходиться).

Так званий топографічний кретинізм - не міф, що не модний термін і не хвороба, а результат недостатньо розвиненого просторового мислення. Щоб ваша дитина не «блукав в трьох соснах» з незavidною постійністю, тренуйте об'ємне мислення з самого дитинства!

**Успішний у спорті.** Для багатьох видів спорту (баскетбол, теніс, футбол, бойові єдиноборства та ін.) Потрібен хороший окомір, вміння точно і швидко оцінити відстань і взаємне розташування в просторі об'єктів - і відповідно розрахувати свої сили.

**Знає, де у нього що лежить.** Люди з розвиненим об'ємним мисленням вміють раціонально організувати свій життєвий простір. Дитина знає, де кожна конкретна річ - в кімнаті або в шкільному ранці.

**Отримує хороші оцінки з малювання та праці.** Тривимірне мислення допомагає достовірно зображати предмети на папері (малювання, креслення, аплікація) і у вигляді об'ємних виробів.

**Часто виграє у однолітків в шахи, шашки, ...а також комп'ютерні ігри (дії-стрілялки), тетріс - в результаті користується заслуженим авторитетом.**

**Охоче вчиться.** Просторове мислення допомагає представляти процеси, дії, предмети наочно, в тому числі в динаміці. Завдяки візуалізації процес будь-якого навчання стає цікавим, а отже - більш ефективним.

**Любить читати.** Або хоча б відноситься до читання за шкільною програмою без гострої ненависті. Тривимірне мислення допомагає представляти сюжет, як наяву (або як в кіно), і літературні твори краще засвоюються.



*Сконцентрований, уважний і добре запам'ятовує інформацію. Це приємні побічні ефекти розвиненого об'ємного мислення (рис. 2.5.).*



Рис. 2.5. Розвинене просторове мислення

*Успішно вирішує завдання з математики, особливо з геометричним змістом. Надалі у нього не буде труднощів і з кресленням, фізикою і астрономією. Хімія, біологія - теж з більшою ймовірністю підуть легко.*

Тепер можна бути впевненим, у вашої дитини не буде проблем з просторовим мисленням.

## **2.2. Зв'язок просторового мислення з іншими здібностями людини**

### ***Розвиток просторової уяви і логічного мислення***

Логічне мислення і просторову уяву відносять до пізнавальних процесів,

також, як і увага, пам'ять, сприйняття і мова.

Коли говорять про загальні здібності людини, то мають на увазі рівень розвитку його пізнавальних процесів: чим краще розвинені у людини ці процеси, тим більш здатним він є, тим більшими можливостями він володіє. Від рівня розвитку пізнавальних процесів учня залежить легкість і ефективність його вчення.

Пізнавальні процеси (сприйняття, увага, пам'ять, мислення) входять як складова частина в будь-яку людську діяльність і забезпечують ту чи іншу її ефективність. Пізнавальні процеси дозволяють людині заздалегідь намічати цілі, плани і зміст майбутньої діяльності, програвати в розумі хід цієї діяльності, свої дії і свою поведінку, передбачати результати своїх дій і керувати ними в міру їх виконання.

Крім загальних здібностей людина може володіти і спеціальними здібностями, наприклад, математичними.

В даний час в країні виникла гостра необхідність в людях, які виявляють здібності у відповідній області, області математики.

Всім добре відомо, що основний внесок в розвиток тієї чи іншої науки роблять математики. Все це висуває перед школою завдання всебічного розвитку в учнів математичних здібностей, завдання підвищення рівня математичної культури, рівня математичного розвитку школярів.

Можливо, багато хто скаже, що не у всіх дітей є математичні здібності.

Вченими встановлено, що кожна здорова дитина має здібності, достатні для засвоєння всіх предметів шкільної програми (рис 2.6).

Ось чому можна стверджувати, що певний рівень математичних здібностей властивий кожному школяреві. Необхідно тільки розуміти,

що існують рівні їх розвитку, педагогу необхідно вміти їх виявляти та розвивати.

Дитячий мозок дуже пластичний і має величезні можливості і резерви. При правильній організації роботи з дітьми, можна домогтися дуже хороших результатів. Все залежить від старанності і наполегливості в досягненні мети.

Ефективність вивчення математики особливо залежить від рівня розвитку певних пізнавальних процесів людини.



Рис. 2.6. Дитина й математика

*Отже, на розвиток яких пізнавальних процесів слід звертати особливу увагу?*

Вчені В. Хаєкер і Т. Циген виділили чотири основних складних компонента, що становлять "ядро" математичного мислення: просторовий, логічний, числовий і символічний. Далі вони розклали ці компоненти на більш прості складові. Вийшла така схема

## 1. Просторовий компонент.

- Розуміння просторових фігур, образів і їх комплексів (синтезів).
- Пам'ять на просторові образи (просторові уявлення).
- Просторові абстракції (вміння бачити у просторових об'єктів загальні ознаки).
- Просторове комбінування (розуміння і самостійне знаходження зв'язків і відносин просторових об'єктів).

## 2. Логічний компонент.

- Створення понять і понятійних абстракцій.
- Розуміння, запам'ятовування і самостійне знаходження спільних понятійних зв'язків.
- Розуміння, запам'ятовування і самостійне виведення висновків і доказів за правилами формальної логіки.

## 3. Числовий компонент.

- Створення числових уявлень.
- Пам'ять на числа, числові рішення.

## 4. Символічний компонент.

- Розуміння символів.
- Запам'ятовування символів.
- Операції з символами.

Для цілеспрямованої та успішної роботи з розвитку математичних здібностей, необхідно проводити роботу по розвитку просторової уяви, комбінаторних здібностей, об'ємно-просторового мислення і логічних операцій мислення: аналіз, синтез, порівняння, узагальнення, класифікація.

Хочеться особливо сказати про уяві. Яку функцію в житті людини вона виконує.

### 2.3. Розвиток уяви

Уява - це особлива форма людської психіки, що стоїть окремо від інших психічних процесів і разом з тим займає проміжне положення між сприйняттям, мисленням і пам'яттю. Уява значно розширює і поглиблює процес пізнання об'єктивного світу.

У житті людини уява виконує ряд специфічних функцій. Перша з них полягає в тому, щоб *представити дійсність в образах*.

Друга функція уяви полягає в *регулюванні емоційних станів*.

Третя функція уяви пов'язана з його участю в *довільній регуляції пізнавальних процесів*. За допомогою майстерно викликаних образів, людина може звертати увагу на потрібні події.

Четверта функція уяви полягає в *формуванні внутрішнього плану дій* - здібності виконувати їх в думці, маніпулюючи образами.

П'ята функція - *це планування і програмування діяльності*, складання таких програм, оцінка їх правильності, процесу реалізації.

За допомогою спеціальних прийомів і вправ можна розвивати уяву.

## **Розвиток просторового та творчої мислення**

Просторова уява - це вміння подумки моделювати, визначати співвідношення між окремими елементами зображення, подумки змінювати їх взаємне розташування, розчленовувати фігуру на частини або «склеювати» її з наявних частин, «представляти» різні конструкції, бачити їх внутрішнім зором в кольорі і деталях.

Використання на заняттях геометричних конструкторів створює оптимальні умови для розвитку просторового мислення і творчої уяви, мислення, кмітливості, фантазії, творчих і комбінаторних здібностей молодших школярів. Складаючи різні силуети, діти засвоюють способи з'єднання елементів, вчать поєднувати їх за розміром, співвідношенням сторін, що сприяє розвитку окоміру і комбінаторних здібностей.

## **Розвиток комбінаторних здібностей, просторового мислення**

Під впливом конструктивної діяльності у дітей розвиваються складні види зорового аналізу і синтезу, включаючи здатність подумки розчленовувати сприймається об'єкт на частини в зоровому полі, досліджуючи кожну з них окремо і потім об'єднуючи в єдине ціле (рис 2. 7.).

Комбінаторні здібності характеризують здатність до виявлення різного роду зв'язків, співвідношень і закономірностей. У широкому сенсі слова - це здатність комбінувати в різних поєднаннях (просторово-часових, причинно-наслідкових, категоріально-змістовних) елементи проблемної ситуації і власних знань.

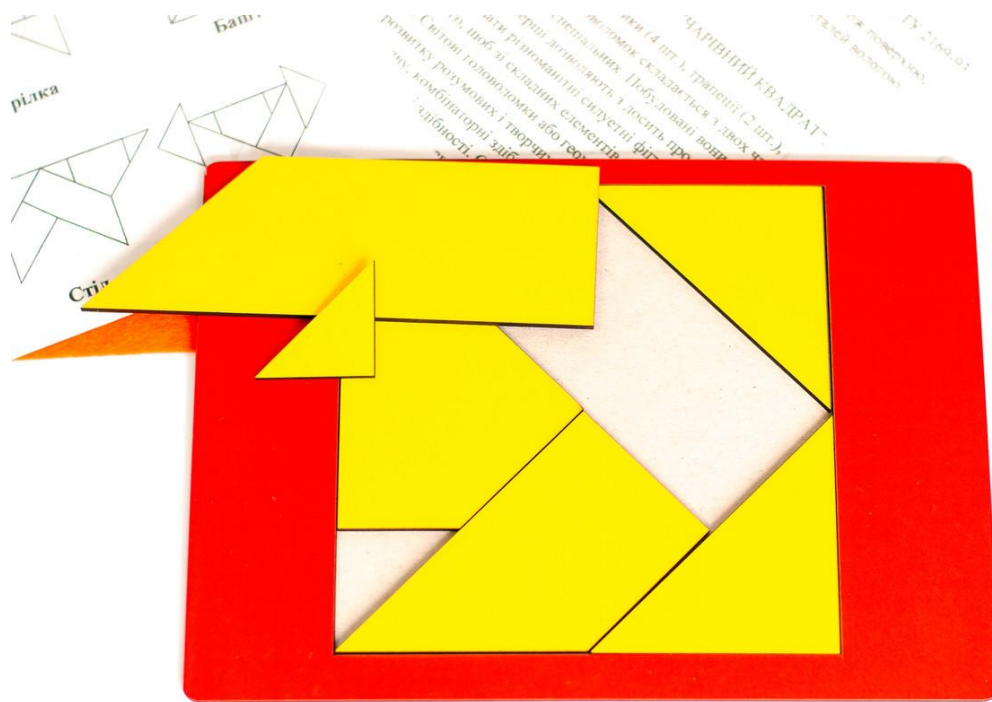


Рис. 2.7. Комбінація фігур в одне ціле

### **Розвиток об'ємно-просторового мислення**

Об'ємно-просторове мислення - здатність створювати образи в тривимірному просторі подумки представляти необхідну конфігурацію, утримувати в зоровому полі відразу кілька об'єктів і оперувати ними.

Уміння створювати просторовий образ і оперувати його становищем - це основа орієнтації в просторі, це необхідні вміння для успішного навчання основам багатьох наук. Вони мають велике значення в житті, тому вимагають цілеспрямованого розвитку.

### **Розвиток логічних операцій мислення: аналіз, синтез, порівняння, узагальнення, класифікація.**

Роль логічного мислення у формуванні математичних здібностей очевидна.

Мислення є вищим пізнавальним процесом.

Відмінність мислення від інших психологічних процесів полягає в тому, що воно майже завжди пов'язане з наявністю проблемної ситуації, завдання, яку потрібно вирішити, і активною зміною умов, в яких ця задача задана.

Розвиток логічного мислення залежить від сформованості і розвитку основних операцій мислення: аналізу, синтезу, порівняння, узагальнення, класифікації, абстрагування, конкретизації.

Кожній людині важливо навчитися зіставляти, порівнювати, аналізувати, знаходити приватне і загальне, узагальнювати по одному або декількох підставах, класифікувати, виділяти ознаки предметів, явищ, давати обґрунтоване доказ, розуміти причинно-наслідкові зв'язки, встановлювати прості закономірності, чітко висловлювати свої думки.

Включення в програму логічних задач, а також різноманітних завдань на формування логічних суджень, збудованих в цільну систему, і методично правильно побудована робота по їх вирішенню надає сприятливі можливості для розвитку розумової діяльності учнів.

Розвиток математичних здібностей безсумнівно залежить від рівня розвитку пізнавальних процесів, зазначених вище. Тому дуже важливо організувати систематичні додаткові заняття з дітьми з розвитку пізнавальних здібностей в позаурочний час, виконувати певні завдання, тренуватися над розвитком цих процесів.

### **Як це все допоможе батькам?**

Розвиток об'ємного мислення відбувається поетапно. Уявлення про тривимірності повинні накопичуватися поступово. Подавайте будь-яку нову інформацію легко, ненав'язливо, без примусу - і тільки після



засвоєння попередніх понять і активного використання щойно придбаних навичок. Пам'ятайте про універсальний навчальний принцип «від простого - до складного» (рис. 2.8.).



Рис. 2.8. Розвиток дитини

**Розуміння вікових етапів розвитку просторового мислення  
допоможе батькам:**

- не поспішати, не форсувати вивчення аспектів, тимчасово складних для засвоєння;
- визначити контрольні точки - якісь орієнтири, підказки про те, чому саме в різному віці навчати дитину;
- зберігати спокій в разі, якщо здається, що дитина «не тягне», «відстає», не справляється з деякими завданнями на оцінку просторового мислення;
- підготувати дитину до успішного освоєння шкільної програми.

До підліткового віку розвиток тривимірного мислення вважається повністю завершеним, але навіть в дорослому віці має сенс тренувати його за допомогою вправ і вирішення просторових задач.

Професор психології Девід Утта з Північно-Західного Університету (США) і професор психології Нора Ньюкомб з Університету Темпл (США) науково довели, що люди різного віку завдяки навчанню можуть поліпшити практично всі відомі типи просторових навичок.

*«Вчитися і розвивати зорові і незорові просторові здібності ніколи не рано»*

Чудовим доповненням до шкільних завдань на тривимірне мислення стане онлайн-платформа із завданнями по цікавій геометрії і вправами на розвиток логіки і мислення в цілому.

Вивчайте теорію, виконуйте і вирішуйте:

- « Завдання на перестановки »
- « Завдання на шаховому полі »
- « Фігури. Розгортки. Відображення »
- « Периметр і площа »
- « Алгоритми »
- «Шестерінки»
- «Стільники. Судоку. Японські кросворди »

*«Перераховані навчальні блоки безпосередньо пов'язані з орієнтацією в просторі і безпосередньо допомагають розвивати об'ємне мислення»*

Дошкільнятам іноді складно виокремлювати окремі ознаки в ряду об'єктів - наприклад, тільки форму, розмір або тільки місце розташування. Тому важливо навчити дитину розділяти просторові та інші ознаки

сприйняття предметів окремо і в групі. Допоможуть завдання на виявлення закономірностей, такі як «Знайди зайву фігуру» (рис. 2.9.) або «Розгадай закономірність і визнач-яке зображення повинно бути наступним».



Рис. 2.9. Гра «Знайди зайву фігуру»

## 2.4. Приклади завдань для розвитку просторового мислення

*Починати роботу потрібно вже з дошкільнятами.*

### Завдання для дошкільнят

1. Завдання з розвитку зорово-рухової координації.

2. Якщо у дитини не сформована ЗРК, йому важко орієнтуватися в двомірному просторі, в просторі зошиту, на столі, аркуші паперу, в книзі і т. д.
3. Завдання з розвитку наочно-образного мислення.
4. Завдання і вправи на розвиток міжпівкульної взаємодії.
5. Завдання з розвитку здатності до комбінування.

### **Завдання для учнів 1 класу**

1. Продовження роботи з розвитку ЗРК, міжпівкульна взаємодії.
2. Завдання з розвитку внутрішнього плану дій.

Якщо у дитини несформована ВПД (внутрішній план дії) він не зможе виконувати усні обчислення без наочної опори.

3. Завдання на розвиток здатності до комбінування.
4. Завдання з розвитку наочно-схематичного мислення.
5. Завдання з розвитку вміння складати різні комбінації з числами, фігурами.

Хоча багато подібних завдання для дітей представляють труднощі, вони з великим інтересом виконують дані завдання. Діти інтуїтивно відчують, що сприяє їх власного розвитку.

Відомо, що будь-який розвиток - це подолання труднощів.

### **Завдання для учнів 2 класу**

1. Обов'язкове продовження роботи з розвитку зорово-рухової координації. Часто зустрічає помилка, що вже ЗРК сформовано і вже

немає потреби виконувати подібні завдання. В тому то і справа, що в другому класі обов'язково повинні бути завдання на розвиток ЗРК. Але малюнки повинні бути більш складні (косі лінії, лінії по діагоналі, «вишивка», дзеркальне відображення малюнка по клітинках). Про необхідність їх виконання ви можете переконатися, прочитавши літературу по психології, а ще простіше, дати подібні завдання дітям. Діти виступають тут як індикатори, якщо вони виконують подібні завдання з інтересом, значить їм це ще потрібно, для їх же розвитку. Діти інтуїтивно відчують, що сприяє їх власного розвитку.

2. Розвиток оперативної пам'яті.
3. Рішення логічних, нестандартних завдань.
4. Завдання з розвитку словесно-логічного мислення.
5. Більш складні завдання з розвитку ВПД. Повороти фігур мислення за годинниковою стрілкою і проти.

Подібні завдання діти виконують з цікавістю. А інтерес - найкраща з мотивацій, він робить дітей по-справжньому творчими особистостями і дає їм можливість відчувати задоволення від інтелектуальних занять.

### **Завдання для учнів 3-5 класів**

1. Завдання з розвитку об'ємно-просторового мислення.

Добре відомо: чим вище рівень просторового уявлення учнів, тим простіше навчати їх геометрії, тим цікавіші завдання можна ставити перед ними.

2. Завдання з розвитку логічних операцій мислення.

3. Завдання на розрізування, перекроювання і складання фігур.

4. Побудова розгорток.

Всі представлені завдання будуть дуже корисні для розвитку здібностей дітей.

Найголовніше заняття необхідно будувати так, щоб у дітей завжди виникало відчуття цікавості. **Що ж буде далі? Цікавість породжує** пошук, визначає і спрямовує його, але саме дослідження призводить до навчання. Знання та навички, набуті таким чином, зберігаються надовго, тому що вони повністю належать дитині. Він їх пережив, а не вивів в результаті логічних умовиводів. Такий спосіб навчання дуже важливий для дитини, оскільки призводить до безперервних змін та особистісному зростанню.

Захопившись завданнями, дитина навіть не думає про те, що він вирішує складну і важливу задачу.

Осі декілька загальних ідей для розвитку просторового мислення дитини:

- Складайте з дитиною фігурки в техніці оригамі, збирайте пазли (в тому числі тривимірні (рис. 2.10.)), ліпіть з пластиліну, вирізуйте з паперу симетричні предмети або сніжинки, грайте в настільні ігри (танграм, шашки, шахи, нарди, «дії», стратегії, Дженга). Малюйте з дитиною плани - місцевості, приміщення.
- Розгадуйте загадки: вони вчать дитину подумки відтворювати предмет по його ознакам, тим самим вибудовувати образ.
- Поясніть дитині як легко визначити сторони світу біля вашого будинку: де сходить сонце, де ви спостерігали захід. Гуляєте пішки в знайомих місцях - запропонуйте представляти, куди ви прямуєте

(північ, південь, захід, схід).

- Виходьте на вулицю - запитуйте думку дитини, як краще дістатися до потрібного місця.
- Дозволяйте допомагати вам збиратися в дорогу. Валіза, в якій не поміщаються речі, - відмінний тренажер тривимірного мислення.
- Ховайте «скарби» і пропонуйте знаходити їх за картками, схемами.
- Оптимальний підхід - комбіновані заняття у формі гри. Тренуйте просторове мислення «на місцевості» та виконуйте спеціальні розвиваючі завдання.

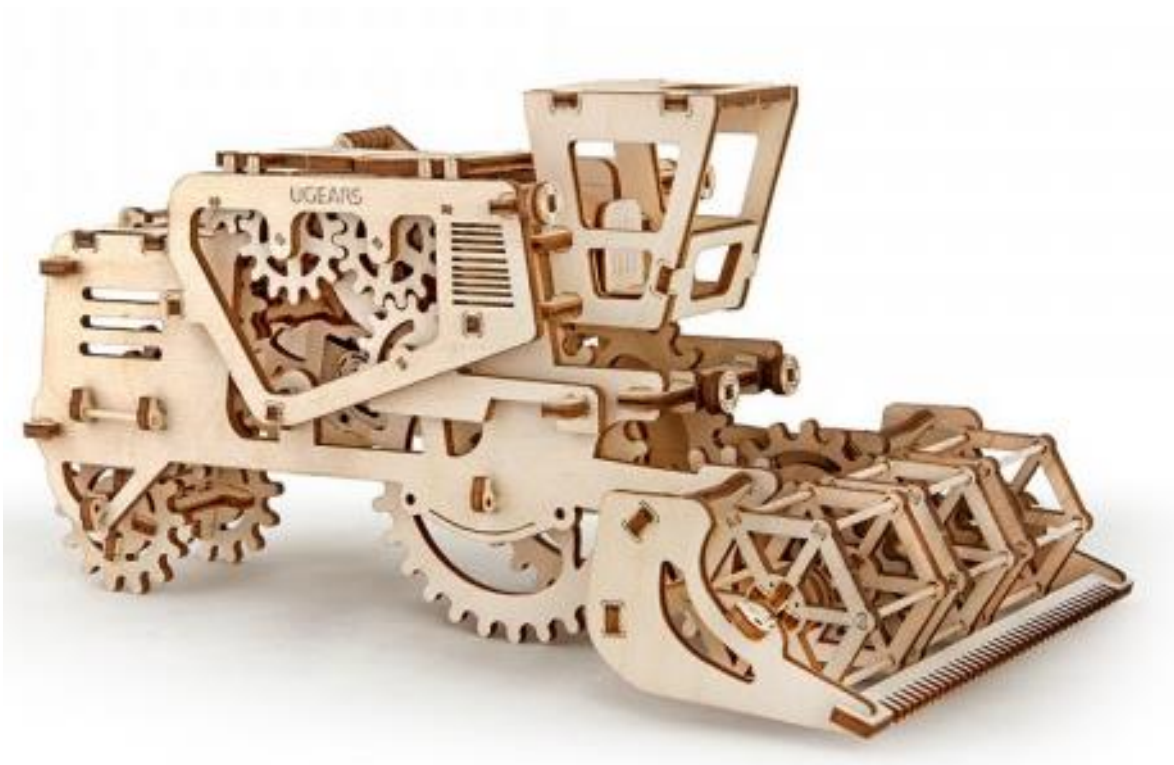


Рис. 2.10. Дерев'яний 3Д пазл

Для більш молодших дітей рекомендуються такі ігри та задачі:

- малювати фігури з точок (повторюючи зразок);
- повторювати візерунки, орнаменти і фігури, що складаються з ліній;
- порівнювати фігури за розміром і формою;

- домальовувати другу (точно таку ж, але в дзеркальному відображенні) половинку картинки;
- писати математичні диктанти;
- з'єднувати точки по цифрам;
- перемальовувати фігури, вносячи зміни (наприклад, збільшуючи або зменшуючи розмір, додаючи деталі), - або по пам'яті;
- дзеркально відображати будь-які малюнки, літери, цифри;
- малювати по інструкції ( «намалюй маленький квадрат в правому верхньому куті листа»);
- проходити лабіринти ( «допоможи зайцю дістатися до морквини, але не потрапити в лапи до вовка»);
- грати з рамками-вкладишами;
- складати впізнавані предмети (кішку, будиночок, вантажівка) з геометричних фігур різного розміру;
- малювати креслення і схеми різних предметів;
- описувати на картинках розташування героїв і предметів, використовуючи прийменники (в, над, під, між, у);
- майструвати вироби, в тому числі об'ємні, - вимагають активного вимірювання і співвіднесення частин цілого.

*А.Л. Венгер пропонує стимулювати дитяче тривимірне мислення за допомогою завдання «Машина їде по дорозі». [11]*

На аркуші паперу намальована дорога: вона може бути прямою і широкою (простий варіант для «початківців») або звивистою і вузькою. У одного кінця дороги намальована машина, в іншого - будиночок. Дитина повинна, не відриваючи олівця або ручки від паперу, «провезти» машину до будинку (намалювати її траєкторію). Замість машини може бути



метелик, замість будинку - квітка, а замість дороги - траєкторія польоту метелика, зображена курсивом.

---

*А.Е. Симановский придумав для розвитку дитячого об'ємного мислення вправа «Влуч олівцем в гуртки». [11]*

Зобразіть на аркуші паперу (або знайдіть в інтернеті шаблон і роздрукуйте) ряди кіл діаметром 3 мм (для початку можна зробити більші кола), розташуйте їх п'ятьма рядами по 5 кіл в ряду. Відстань між колами повинно бути однаковим, наприклад, 1 см. Дитині потрібно, не відриваючи передпліччя від столу, як можна точніше і швидше розставити крапки в усі кола. При цьому обговоріть «методику»: або точки потрібно проставляти зліва направо в першому рядку, справа наліво в другій і т.д; або нехай дитина ставить крапки зверху вниз в першому стовпці, від низу до верху в другій і т.д.

### **Прості ігри на розвиток просторового мислення**

#### *Гра «Поклади кішку»*

Запропонуйте дитині уявити якусь тварину (кішку, слона, таргана) - нехай відповість, в якій предмет ця тварина вміститься. У склянку? А в коробку від телевізора? Може, в проїжджаючий повз вантажівка?

#### *Гра «Розставити правильно!»*

Дайте дитині різні предмети і запропонуйте помістити-розставити їх, слідуючи вашим вказівкам: ближче (ближче ніж), далі (далі ніж), трохи вперед, назад, ліворуч від і т.п.

### ***Гра «Метаморфози»***

Попросіть дитину намалювати коло за квадратом, трикутник перед прямокутником. Можна ускладнити завдання: намалювати циліндр перед кубиком або зобразити будиночок з іншого боку, зверху, «розрізати» іграшкові гантелі навпіл і т.д.

### ***Гра «Спецагент на завданні»***

Запропонуйте дитині уважно оглянути приміщення, де він знаходиться, і запам'ятати все довкола. Потім задавайте питання, використовуючи слова - покажчики розташування: якого кольору стіл стоїть зліва від тебе? який предмет знаходиться прямо під люстрою? Ця гра ще цікавіше на вулиці - там вже можна задіяти рухомі предмети.

### ***Гра «Малюнок на спині»***

Малюйте у дитини на спині різні фігури, потім предмети - нехай намагається вгадати, що ви вдаєте (рис. 2.11.).



Рис. 2.11. Гра «Малюнок на спині»

### ***Гра «Муха»***

У неї грають дві людини, плюс один «спостерігач». Гравці представляють - а спостерігач розкреслює - ігрове поле: гра розміром 9 квадратів в довжину і 9 в ширину. У лівому верхньому кутку потрібно подумки помістити муху. Гравці по черзі роблять ходи, переставляючи муху на різні квадрати, а спостерігач зазначає ці ходи на ігровому полі. Коли спостерігач зупиняє гру, кожен гравець називає квадрат, в якому, на його думку, знаходиться муха. Перемагає той, хто назве вірно.

### ***Гра «Маршрут руху»***

Намалюйте складну схему вулиць міста (або просто візьміть карту). Поясніть дитині, що він - поліцейський, який виїхав з пункту А (покажіть його на схемі) в пункт Б (теж покажіть). Дитина повинна проїхати по маршруту, називаючи кожен поворот. При цьому варіанті гри активними словами будуть «направо» і «наліво». Потім «змініть роль»: тепер дитина - диспетчер, який відстежує траєкторію на карті. В описі маршруту повинні з'явитися слова «вгору» і «вниз».

### ***Гра «Чарівний мішечок»***

Складіть в невеликий мішечок різні фігурки - краще стереометричні (кубик, кулька і ін.), Але можна і ігрові (пірамідки, матрьошки і ін.). Запропонуйте дитині на дотик вгадати, що в мішечку.

## ВИСНОВОК ДО РОЗДІЛУ 2

Було розглянуто багато методик для розвитку просторового мислення людини. Дивлячись на них можемо зробити висновок, що найвдаліший час для розвитку цього вміння це вік до 12 років, адже саме в цей період дитини отримують усі базові навички для життя, а надалі лише розвивають їх. І несерйозне відношення до розвитку просторового мислення, погано скажеться в подальшому.

Найголовніше, заняття необхідно будувати так, щоб у дітей завжди виникало відчуття цікавості. Що ж буде далі? Цікавість породжує пошук, визначає і спрямовує його, але саме дослідження призводить до навчання. Знання та навички, набуті таким чином, зберігаються надовго, тому що вони повністю належать дитині. Він їх пережив, а не вивів в результаті логічних умовиводів.

## РОЗДІЛ 3. Приклад відеоігор для розвитку просторового мислення

Значна частина всіх додатків, які щодня з'являються в електронних магазинах, намагається щось розвивати в своїх користувачів, щоб зробити їх краще і розумніше. Тепер можна просто завантажити сотні програм для розвитку свого мозку. Але всі вони настільки нудні і стомлюючі, що можуть розвинути хіба що нудьгу. З іншого боку, постійно з'являється величезна кількість нових ігор. Але більшість з них виконує виключно розважальну функцію. Тому я вирішив знайти ігри які можуть об'єднати цікаве з корисним. Сьогодні мова піде про ігри, які замість звичайного геймерського фана дають можливість розвивати свій мозок і дозволяють прокачати увагу, просторове мислення та інший корисний функціонал свого мозку[7].

### 3.1. Thomas Was Alone



Рис. 3.1. Рівень з гри

Кафедра КІТ				НАУ 20 26 27 000 ПЗ			
<i>Виконав</i>	<i>Семенов Н.М.</i>			Приклад відеоігор для розвитку просторового мислення	<i>Літера</i>	<i>аркуш</i>	<i>аркушів</i>
<i>Керівник</i>	<i>Холявкіна Т.В.</i>					53	17
<i>Консульт.</i>					УС 211М 122		
<i>Н. контроль</i>	<i>Райчев І.Е.</i>						

### *Коротко*

Геометрична гра, що розповідає про пригоди маленького прямокутника по імені Томас в мінімалістичному світі гострокутих лабіринтів. Мета ігрового процесу зводиться до того, щоб провести Томаса з точки А в точку Б через низку складних перешкод і головоломок. В процесі подорожі він зустрине багато друзів різної форми, які будуть допомагати йому вирішувати різні завдання, а все це буде дотепно коментувати приємний закадровий голос англійського коміка Денні Уоллеса.

Найцікавіше, що за цими геометричними викрутасами в результаті виявляється справжнісінька людське життя з усіма складнощами взаємовідносин: любов, дружба, лідерство - саме це за задумом творця Thomas Was Alone Майка Байселла і є ключовою частиною гри. При цьому її складність коливається між «дуже просто» і «хм, як же так». Також, якщо вам сподобається знущатися над своїм мозком, вирішуючи логічні задачі по переміщенню героя з одного місця в інше, можете спробувати Limbo - правда, особливо вразливим краще триматися від неї подалі.

### *Детально*

Баги, глюки, періодичні збої комп'ютерних програм. Навряд чи зараз когось можна цим сильно здивувати. Як би далеко не зайшов технічний прогрес, цим поняттям завжди буде місце в сфері цифрових технологій. Програмісти усіма силами намагаються боротися з ними, а споживачі електронних пристроїв змушені лише сподіватися на те, що коли-небудь їх не стане ... Мабуть, ніхто і ніколи не замислювався: а як же бути збоєм комп'ютерної програми? Першим це зробив якийсь Mike Bithell у своїй новій грі Thomas Was Alone, яка, можливо, зможе змінити ваше ставлення до цих пережитків сучасних технологій. Ні рук, ні ніг, ні голови.

Лише помаранчевий квадрат на фоні таких же паралельно-

перпендикулярних текстур. Знайомтеся - Thomas-AT-23-6-12 - штучний інтелект, що з'явився в потрібний час в потрібному місці. Збір комп'ютерної програми породив на світло помилку, яка була покликана не відрізнитися від мільйона подібних їй. Томас і є звичайна помилка, якби не одна обставина: він здатний спостерігати за оточенням і аналізувати те, що навколо нього.

Іншими словами, Томас здатний думати. Він опинився абсолютно один в цьому світі і, здавалося, ніщо не може перемогти його самотність. Так почалася історія звичайної комп'ютерної помилки. Так почалася історія Томаса.



Рис. 3.2. Другий рівень з відеогри

«Хто ж я? Де я?» - раз у раз він ставив собі ці питання, відповіді на які, можливо, він ніколи не отримає. Здатність до спостереження тепер його єдиний друг. «Треба ж, я навіть можу стрибати» - подумав він, вперше встрибнувши на невеликий уступ перед ним. Світ здається йому небезпечним і в будь-який момент може вбити його. Це не записки божевільного, це думки нашого героя, вимушеного по крупицях збирати інформацію про себе і про навколишній його світ. Він не знає куди він йде і навіщо. Є лише він і його думки. Погодьтеся, зав'язка дуже інтригує

навіть при всій своїй тривіальності.

Власне, на взаємодії Томаса з навколишнім світом і будується весь ігровий процес. Основним завданням гравця є провести Томаса через певні перешкоди до таємничого порталу, використовуючи лише одну його здатність - стрибати. Все це супроводжується спокійною розслаблюючою музикою і оповідальними коментарями диктора, який і розповідає гравцеві все, що відбувається з Томасом і про що той думає. У міру просування по сюжету Томас зустрічає інші помилки, які також мають ім'я і, більш того, свій неповторний характер. Знайомство з новими AI, з одного боку, досить радує Томаса, з іншого - засмучує, адже кожна з помилок поводить ся і думає практично як людина, а людина схильна до різного роду емоціям і вчинків. Тепер він начебто не самотній, але чи так це насправді?



Рис. 3.3. Приклад взаємодії у грі

Поступово помилки починають пристосовуватися один до одного і допомагати в силу своїх здібностей. Наприклад, Кріс - маленький світло-коричневий квадрат, на відміну від Томаса, стрибає вельми невисоко, але



здатний пролізти туди, куди інші не зможуть, або просто допомогти іншим застрибнути. Джон - жовтий вертикальний прямокутник, вельми стрибучий і може забратися туди, куди іншим не під силу. Клер - великий синій квадрат, вміє плавати і може перевозити на собі інших товаришів по нещастю. Сара - рожевий горизонтальний прямокутник, який виступає для інших в ролі батута. Джеймс - зелений квадрат, на який сили тяжіння цифрового світу також діють, тільки з точністю до навпаки. Пересувається він виключно по стелі. Але це ще не все герої, з якими Томасу належить познайомитися на своєму шляху до невідомості.

Зрозуміло що, всіма цими персонажами належить управляти гравцеві, а перемикання між ними відбувається простим натисканням кнопок. Лише грамотно використовуючи можливості кожного з них, можна розраховувати на успіх. Завдання все та ж: провести всіх персонажів до заповітних порталів, які ідентичні розміром кожного з героїв. Помилитися тут практично неможливо, можна просто не дійти до фінішу. Гра поступово набирає обертів, а гравцеві лише залишається робити свою справу і слухати голосу диктора, вустами якого промовляє доля головних героїв. Навіть недосвідченим в жанрі платформера-головоломки гравцям *Thomas Was Alone* навряд чи зможе доставити хоч якісь незручності. Якщо початкові рівні проходяться взагалі без будь-яких проблем, то навіть ближче до кінця гри мінімальний прояв кмітливості допоможе досягти бажаного результату. Десь потрібно використовувати більш стрибучого персонажа, десь того, від якого можна відштовхнутися і підстрибнути ще вище навіть самому непригучему, а десь просто проплисти на старенькій Клер. Адже ось все-таки хороша річ ця взаємовиручка.

Деякі сумніви може викликати лише дисбаланс складності рівнів, який, судячи з усього, живе своїм життям. Деякі з них проходяться лише з кількох спроб, а вже наступний рівень може бути чи не легше початкового

етапу. Ближче до кінця гри етапи ускладнюються лише тим, що деякі місця фарбують героя в той чи інший колір, який відповідає здібностям, властивим даному кольором. Таким чином, за один рівень можна кілька разів поміняти здатності того чи іншого героя. Але, знову ж таки, істотно ці нововведення ні на ігровий процес, ні на складність гри не впливають. Як мінімум, це незвично для подібного роду ігор, де складність зазвичай тільки зростає в міру просування по сюжету. До слова, про сюжет. Безперечно, історія в грі присутній, але багато хто може бути засмучені тим, що їх очікує в кінці. Можливо, розробники не ставили перед собою завдання здивувати всіх сюжетної складової проекту, але все ж історія заслуговує чогось більшого, ніж ми бачимо в грі. Проте, основна ідея Thomas Was Alone реалізована цілком успішно. Дружба і взаємовиручка - ключ до успіху. Як же часом людям не вистачає цього в реальному житті.

А реальне життя наших героїв - це невідоме комп'ютерне простір, яке переповнене різними перегороджують шлях перешкодами. Непереборні виступи, бездонні ями, смертоносні шипи, рови з водою, зникаючі і рухомі платформи - всі ці нехитрі пристосування будуть заважати гравцеві на його шляху до незвіданого. Виглядає все на рівні 8-бітних приставок, тільки набагато краще в графічному плані. Звичайно, картинка не вражає уяву, але для проекту подібного рівня все дуже навіть непогано. Управління персонажами вельме інтуїтивне і кожен з них поводить так, як властиво тільки йому. Навіть до цих кольорових квадратів і прямокутників швидко звикаєш і вони перестають здаватися такими бездушними. Починаєш навіть переживати за них. Мабуть, розробники змогли домогтися того, чого хотіли. Лише допомога один одному може стати надією на порятунок в будь-якій життєвій ситуації, нехай навіть і в віртуальному світі. Чи не це те, над чим нам варто задуматися?

## 3.2. Monument Valley

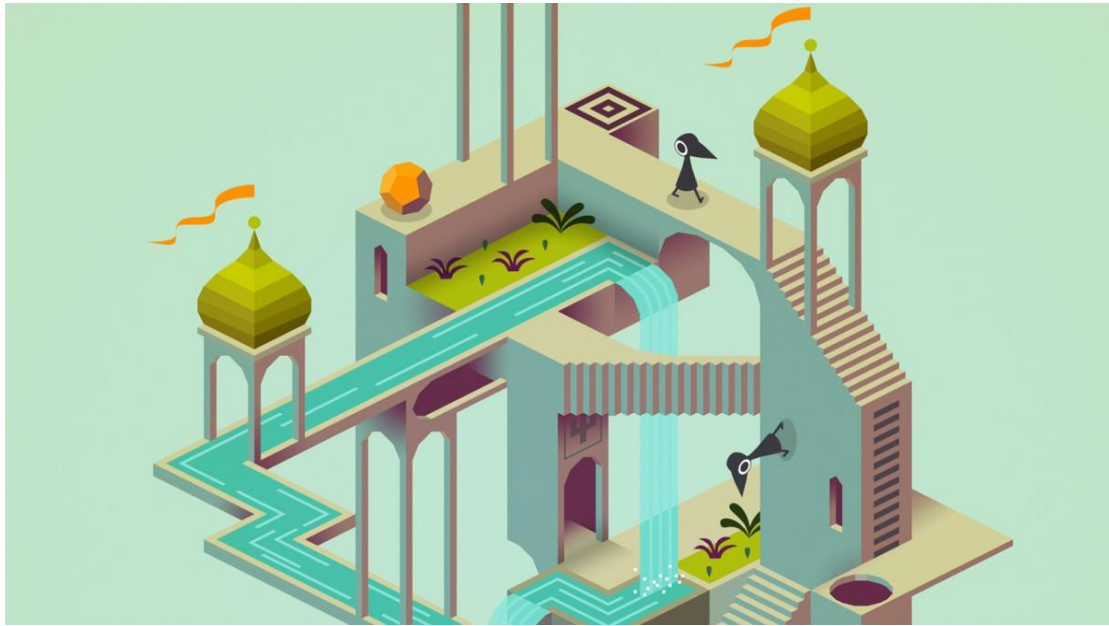


Рис. 3.4. Приклад стилістики рівнів

### *Коротко*

Головний спадкоємець творчості великого художника-графіка Мауріца Ешера і одна з найкрасивіших мобільних ігор сучасності. Гра розповідає про геометричні поневіряння принцеси Іди по архітектурних об'єктах в пошуках пробачення. У Monument Valley вже встигли пограти навіть ті, хто в цілому прохолодно ставиться до будь-яких електронних розваг. І тут вся справа в трьох головних особливостях, які виділяють її на тлі всяких Temple Run і Candy Crush: помірна складність, неймовірна краса і невелика тривалість.

Для непідготовленого користувача Monument Valley може здатися досить складною, але рівно настільки, що рішення всіх її головоломок не розчарує своєю легкістю і в той же час не викличе бажання від злості розбити гаджет об стіну. Вже на третьому рівні мозок починає буквально скипати від нездійсненності поставленого завдання: як же я пройду в ті двері, якщо я ось де, а вона ось там?! Для того, щоб подолати всі десять рівнів, доведеться включити своє просторове мислення і порядком

покриття ігрові пейзажі в уяві. Але незважаючи на гадану нездійсненність, рішення завжди вражаючим чином не примушує себе чекати довше декількох хвилин. При цьому маленька тривалість гри - її спокійно можна пройти менше ніж за півгодини - виступає додатковим мотиватором на проходження: тут же всього десять рівнів, я розберуся з нею ще до заходу сонця і забуду про неї назавжди. Всі ці фактори зі звичайної гри перетворює її в кращий тренажер для розвитку просторового мислення і уяви.

### *Детально*

Кожен рівень в **Monument Valley** - хитромудра головоломка з парадоксальних, «неможливих», фігур. Трикутники Пенроуза (рис. 3.5), сходи Рутерсварда, водоспади, спрямовані в нескінченність и тд. При першому погляді на таку систему може піти обертом, але чим більше в неї вдивляєшся, тим краще починаєш розбиратися в тому, як все це насправді влаштовано. Для ігор використання подібних конструкцій аж ніяк не в новинку. Імп-артом і конкретно роботами великого нідерландського художника Мауріца Корнеліса Ешера надихалися творці чорно-білої **The Bridge**. А японські розробники активно експлуатували принцип «раз намальовано - значить, можна» в мозговиворачіваючій головоломці **Echochrome**.



Рис. 3.5. Трикутник Пенроуза

Але студії Ustwo вдалося не тільки повторити пишність ешерівських конструкцій, а й передати властиву його роботам жвавість, населив світ дивними, але чарівними створіннями, які ходять по стінах, стелях, головою вниз, задом наперед, шкереберть, і плутаються в майже мєбіусовських петлях. А під нашою опікою знаходиться мовчазна принцеса на ім'я Іда. Звідки вона взялася і що їй потрібно - з ходу сказати важко, але на відміну від тієї ж NaissanceE, де розробники спробували обійтися взагалі без слів, в Monument Valley до кожного рівня прикладені простенькі анотації в дусі заходеровського Вінні-Пуха ( «Глава п'ята, в якій Іда зустрічає Слонопотама ... »).

Крім того, іноді який-небудь зустрічний відсуває монологи - загадкові, але лаконічні. У плані подачі історії Monument Valley нагадує Journey : про все повідомляють то алегорично, то, навпаки, подитячому прямолінійно - і завжди по чуть-чуть, але при цьому, якщо напружити уяву, обов'язково складеться загальна картина.

У самій грі вам належить виконати роль провідника - відшукати і прокласти для принцеси шлях від початку і до кінця глави, керуючи простором і деталями неможливою споруди. Вся конструкція знаходиться під нашим контролем; її можна перевертати, крутити, зрушувати, складати, розгортати - трансформувати так, як дозволяють завдання і структура рівня.

Проходиться Monument Valley без особливих зусиль. Всі відповіді приходять поступово, як би самі собою, але при цьому кожен раз дивують. Варто повернути важіль або просто подивитися на лабіринт під іншим кутом, як він дивним чином перетворюється, відкриваючи нові боку і нові можливості. Загадки прості, але розумні і дуже рідко експлуатують уже знайомі ідеї. Десь потрібно повернути рівень так, щоб пара незв'язаних платформ склалася в уявний місток, десь - погратися з розташуванням

блоків і отримати парадоксальну сходи, по якій Іда зможе піднятися (хоча насправді, навпаки, спуститися ... або все- таки ні?) до виходу з рівня. Саме різноманітність і винахідливість розробників (а зовсім не складність) спонукають пройти Monument Valley до кінця.



Рис. 3.6. Головоломка з драбиною

Спокій, умиротворення і тихе благоговіння - ось що ви будете відчувати, проходячи Monument Valley. Вона з першого розділу ненав'язливо стискає в своїх затишних обіймах все сильніше і сильніше. Monument Valley проходиться за один присід, за годину або два, і весь цей час пролітає непомітно. Це майже ідеальна мобільна гра - справжня кишенькова казка. Прикро лише, що обривається вона занадто рано.

### 3.3. Hitman GO

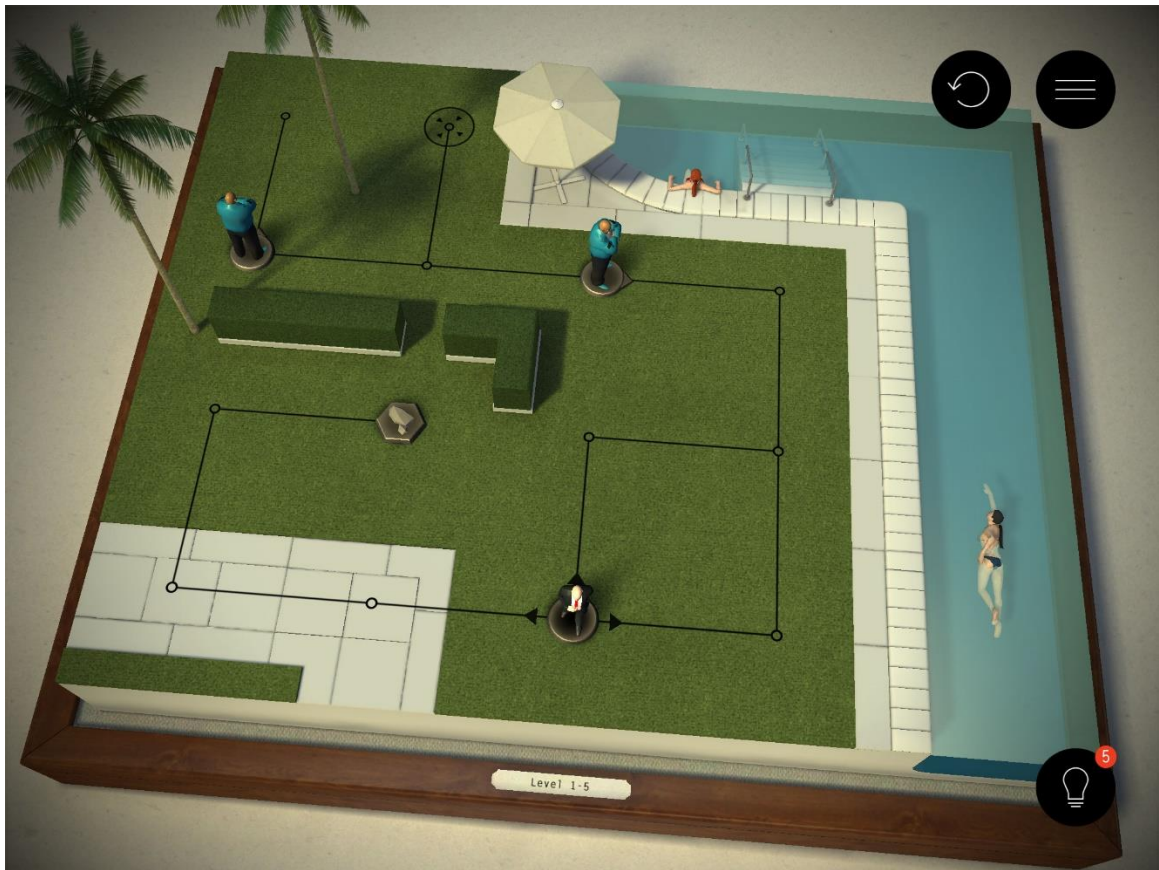


Рис. 3.7. Простий рівень у грі

#### *Коротко*

Квінтесенція стилю, логіки і жорстокості, справжній тренажер просторового мислення для самих холоднокровних користувачів, що цінують естетику найманих вбивць. У цій грі вам буде відведена роль професійного кілера, завданням якого стане усунення важливих осіб і крадіжка секретних даних. Але не переживайте - на ділі все це скоріше схоже на спрощені шахи з майбутнього, ніж на кривавий четвер. Більш того, згідно з досвідом, проведеним автором над самим собою, тривала гра в Hitman GO дуже добре стимулює роботу мозку. Вже на 1/4 ігрового прогресу мозок починає голосно пихкати і перезавантажуватися при спробі

пройти кожен рівень на 100%. А функціонал при цьому продовжує невблаганно розширюватися: з часом у вас, наприклад, з'явиться можливість привертати увагу ворогів киданням каменів або переодягатися, щоб ворожі пішаки спокійно пропускали вас повз. Загалом, щоб вирішити здавалося б просте завдання, доведеться максимально точно планувати всі свої дії і заздалегідь програвати їх у своїй уяві - в іншому випадку вас чекає довгий ворожіння і численні перезавантаження.

### *Детально*

Так, саме так - настільна гра-головоломка. Є поле, на якому розставлені фігури охоронців, що рухаються за заданими правилами. Плюс є додаткові об'єкти: діжки з рослинами за якими можна сховатися, камені, які можна кинути, відволікаючи вартою, одяг для перевдягання, зброю різних типів і т.д. При цьому всі предмети і об'єкти дуже органічно вписані в геймплей і відповідають своїм аналогам з великого Hitman. І, звичайно ж, є мета, яку потрібно усунути або точка виходу, до якої потрібно дістатися. Але загальна схема геймплея Hitman GO не претендує на інноваційність. Дана гра представляє собою нехитру, на перший погляд, «настолку». У ній потрібно пересувати підконтрольну фігурку до заповітної мети, беручи до уваги не надто складні правила. Отже, головний герой - неповторний 47 (точніше, його фігурка). Є ігрова локація, прикрашена декораціями, що називається, «за мотивами». Є розкреслений поле з клітинами, за якими дозволено пересуватися. На цьому полі обов'язково будуть присутні фігурки нехороших дядечок, яких можна або убити (тобто прибрати з поля), або обійти. Перед безмовним кілером ставлять, як правило, кілька завдань. Найважливіша і обов'язкова - пройти від початкової точки до фінальної. Ще на ігровий карті може лежати кейс, забрати який – вже додаткова місія.



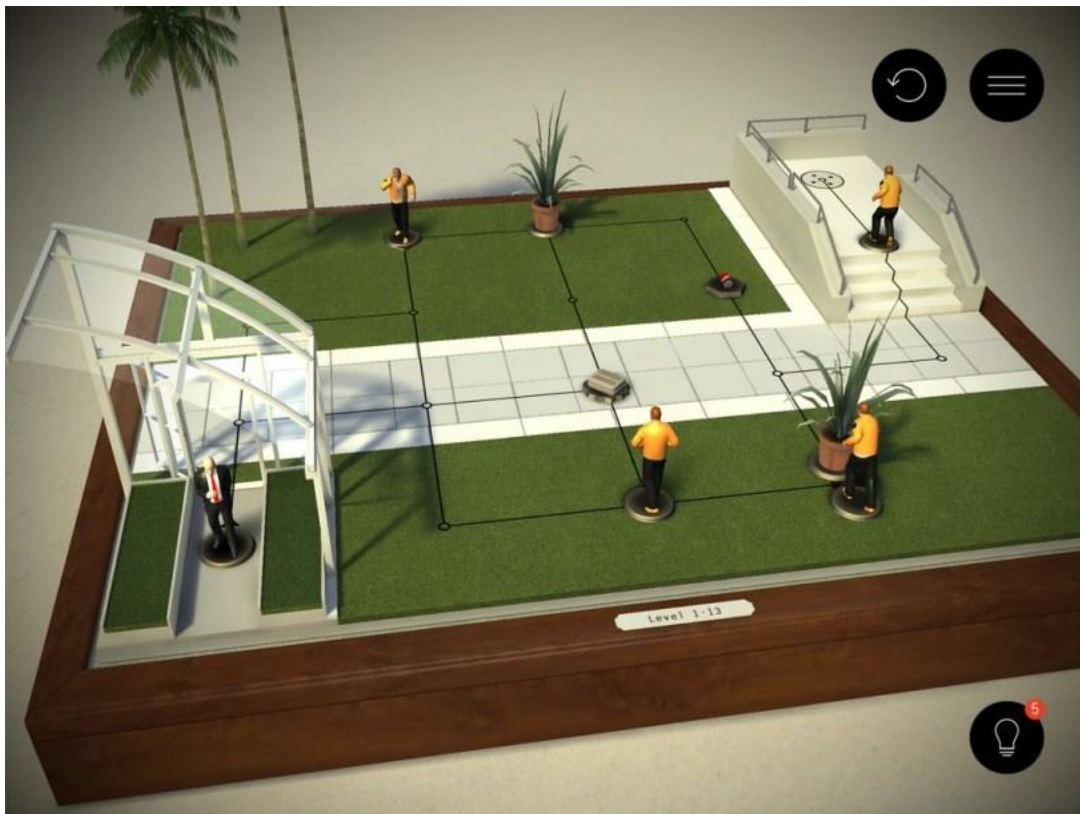


Рис. 3.8. Рівень з іншого боку

Проробляти вищеописані маніпуляції буде не так вже й просто з кількох причин. По-перше, фігурки-охоронці теж вміють пересуватися. Точніше, вони це завжди роблять - крокують по своїй траєкторії на одну клітку за хід. Є нерухомі персонажі, які дивляться в одну точку. Ті ж, хто вміє топати по карті, бачать на одну клітку перед собою - щоб обійти охоронців, потрібно продумувати свої дії на кілька кроків одразу. Зустрічаються і індивіди з собаками (злісні вихованці бачать на дві клітини вперед). Нашому герою є чого протиставити всім скороминущим труднощів. Кілер вміє ховатися в фікусах. Не смійтеся, це дуже допомагає. Також лисий вбивця вміє користуватися телепортами і стріляти з снайперки. Іноді доведеться навіть в переодягання грати, але це вже на більш пізніх рівнях. Вбивати теж дозволительно.

Рівнів в Hitman GO дуже багато, всі вони можуть похвалитися прийнятним різноманітністю. Рішення поставлених завдань потребує

грунтовної продуманості, але варіативності тут, на жаль, немає зовсім. Так, іноді супротивників дозволяється відволікати кинутими каменями. Так, фігурка героя може «топтатися» по двом сусіднім клітинам в очікуванні догляду охоронця на протилежний кінець «дошки». Але це теж є елементом продуманого алгоритму. І навіть коли ви попросите підказки у гри, вона вам намалює лише один варіант для проходження. У будь-якому випадку пошуки такого доставлять любителям настільних забав масу приємних емоцій.

Хоча GO - гра власного формату, вона трохи збиває з пантелику прихильників Hitman: тут взагалі немає ніякої варіативності. Кожен рівень - це підбір єдиного вірного рішення, послідовність заздалегідь продуманих дизайнерами дій. Сенсу проходити завдання повторно і взагалі експериментувати немає - гра все одно не розкриється з нових сторін.

Інша справа, що це досить винахідливий пазл. Як і в великих випусках серії, в GO потрібно продумувати кожен крок і продумувати все наперед - наприклад, відійти за ріг, відвернути охоронця каменем і потім прошмигнути у нього за спиною. Складність зростає в зразковому темпі - тільки ти вирішив одну головоломку, як гра підкидає нову, трохи більш заплутану, або додає на поле новий елемент. Тут немає жартівливо легких рівнів (крім пари вступних), але і вартовими інтелектуальними марафонами гра теж не обертається.

### **3.4. Harmony**

Дивна головоломка, яка пропонує пересувати різнокольорові квадратики відповідно заданої палітри. Затягнувши вас простими завданнями, на кінець першої години harmony почне як слід експлуатувати все півкулі мозку. Згодом кількість відтінків, які потрібно зібрати,

збільшиться від двох до десяти, і з цього моменту гра перестане бути милим тайм-кілером і перетвориться в жорстокого викладача просторового мислення. Зробіть один невірний крок - і партія програна, а вирішите кинути гру - самовдосконалення кінець.



Рис. 3.9. Приклад гри

### Детально

Чим простіше - тим краще. Це твердження вірне для пазл як ніяке інше. А що ще в них можна придумати? Геймплей таких ігор не блищить різноманітністю. Бери та збирай розкидані частини головоломки в єдине ціле. Щоб хоч якось виділитися із загальної маси ігор необхідно придумати щось нове або серйозно модернізувати стару концепцію, що і робить гра har·mo·ny.

Все що від вас буде потрібно зробити - це переміщати плитки по екрану, намагаючись виставити їх у правильній послідовності. Щоб поміняти місцями дві плитки необхідно виділити спочатку одну, а потім тапнуть за іншою, щоб вони помінялися місцями. Зауважте, кількість

можливих переміщень обмежена і відображається невеликою точкою чорного кольору посередині тайтла. Одна точка - одне переміщення, дві точки - два і так далі. Крім того під час гри повинні бути використані всі можливі перестановки. Інтерфейс гри абсолютно плоский і не містить зайвих елементів. Під час гри на екрані відображаються лише кнопка для виходу в головне меню, перемикач рівнів, клавіша для перезапуску поточного рівня з самого початку, кнопка для попереднього перегляду кінцевого зображення, яке, власне, і потрібно скласти, клавіші для скасування останнього скоєного дії і пропуску рівня. До слова, остання кнопка докуповується окремо і обійдеться користувачу в зайві \$ 4.99.

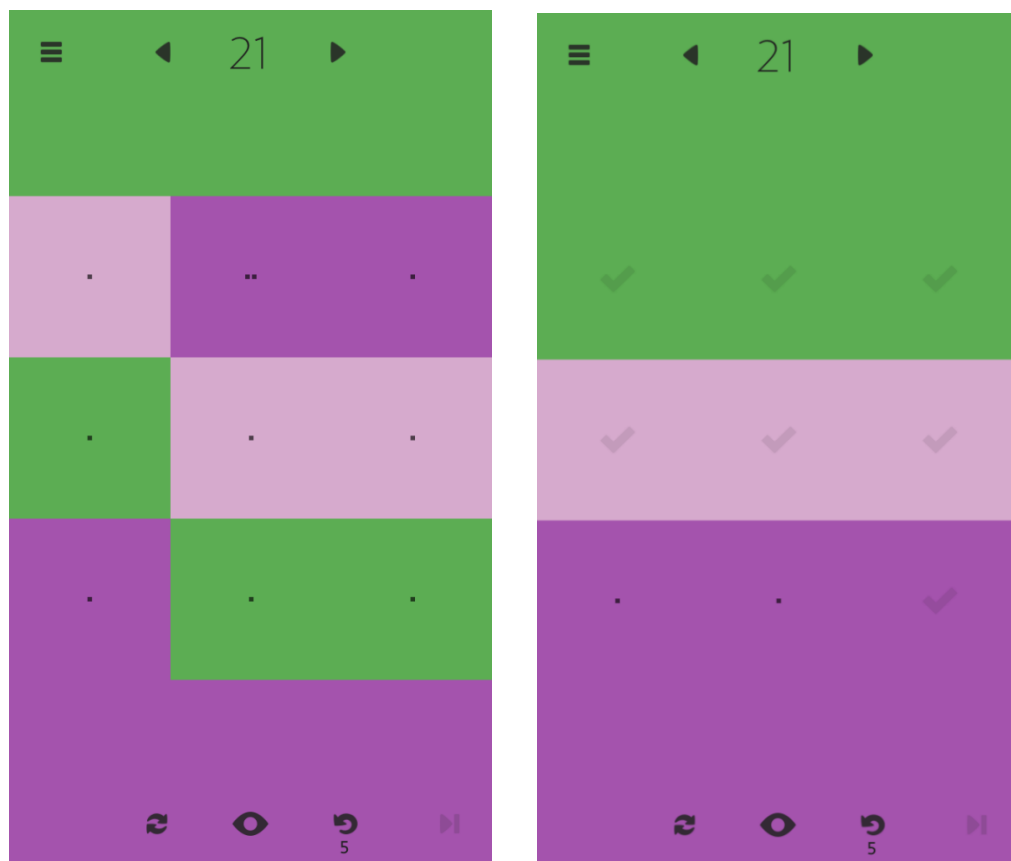


Рис. 3.10. Приклад рівня гри Harmony

Безкоштовних флешбеків всього 5. Решта також купуються за реальну валюту. За 10 штук доведеться викласти \$ 0,99, 50 штук витягнуть з вашого гаманця ще \$ 2,99. За те, що ви поділитесь з друзями в Facebook

інформацією про поточний рівень гри можна отримати зайві 10 флешбеків. Через самої концепції гри необхідності в їх використанні практично немає. Рівні в har • mo • nu змусять вас поламати голову тільки перші хвилини. Після пари пройдених ЛЕВЕЛ ви швидко звикнете. Набагато простіше і дешевше, що важливо, буде почати рівень заново і швидко його пройти. На жаль, саме в цьому криється як позитивна, так і негативна риса гри. har • mo • nu має дуже низький поріг входу і не змусить новачка довго вникати в ігровий процес. З іншого боку, з кожним пройденим рівнем гра швидко набридає. У наступних рівнях змінюється колірна схема, збільшуються розмір сітки плиток і кількість необхідних перестановок. В іншому ж гра не змінюється.

Отже гра har • mo • nu дійсно змушує по-новому поглянути на такий звичний жанр, як пазли. У ній поєднуються непогане музичний супровід, нова концепція і від цього трохи незвичний, але не менш захоплюючий геймплей. Однак підтримати стартовий інтерес гра не в змозі і ефект насичення від неї настає дуже швидко. Купити гру har • mo • nu ви можете в Google Play і Amazon Appstore . Користувачам Apple пощастило трохи більше, так як зараз har • mo • nu поширюється в App Store абсолютно безкоштовно.

### **ВИСНОВОК ДО РОЗДІЛУ 3**

У цьому розділі було наведено багато прикладів відеоігор для розвитку просторового мислення. Кожна з них має свій підхід для вирішення цієї проблеми. І як можна впевнитись вони дійсно є гарним методом для розвитку, адже кожна з них насамперед має у меті розважити гравця, а те що в процесі гри людина розвиває просторове мислення, лише приємний бонус.

Може здатись що тільки дитини краще сприймає інформації в процесі гри, але це не так. Навіть дорослим людям це буде набагато комфортніше та цікавіше вивчати нове, чи розвивати щось конкретне в ігровій формі.

## РОЗДІЛ 4. Вибір типу гри для реалізації

Отже після розбору найвідоміших концептів ігор для розвитку просторового уявлення, я прийшов до висновку що серед них майже немає класичних лабіринтів. Ця давня споруда, а в нашому часі гра, була відома ще у 400 році до нашої ери, завдяки історику Геродоту. Але насправді, з'явилися вони забагато років до цього. Один з найвідоміших лабіринтів по праву можна назвати *Фаюмський лабіринт* [3].

### 4.1. Фаюмський лабіринт

Це найдавніший єгипетський лабіринт (рис. 4.1) знаходився поруч з озером Біркет-Карун, розташованим на захід від річки Ніл, в 80 кілометрах на південь від сучасного міста Каїр. Він був побудований ще в 2300 році до нашої ери і був оточений високою стіною будівлю, де було півтори тисячі наземних і стільки ж підземних приміщень. Лабіринт являє собою надзвичайно велику споруду - розміри його заснування складають 305 на 244 метрів. Греки захоплювалися цим лабіринтом більше, ніж будь-якими іншими єгипетськими будівлями, за винятком пірамід. За часів античності його називали «лабіринт», і він послужив моделлю для лабіринту на Криті.

Над входом в єгипетський лабіринт були написані такі слова: "безумство або смерть - ось що знаходить тут слабкий або порочний, одні лише сильні і добрі знаходять тут життя і безсмертя". Багато легковажних увійшли в ці двері і не вийшли з неї. Це - безодня, яка повертає назад лише сміливих духом.

Кафедра КІТ				НАУ 20 26 27 000 ПЗ			
Виконав	Семенов Н.М.			Вибір типу гри для реалізації	Літера	аркуш	аркушів
Керівник	Холявкіна Т.В.					71	19
Консульт.					УС 211М 122		
Н. контроль	Райчев І.Е.						

Складна система коридорів, дворів і кімнат в лабіринті була такою заплутаною, що без провідника стороння людина ніколи не зміг би знайти в ньому дорогу або вихід. Лабіринт був занурений в абсолютний морок, а коли відкривали деякі двері, то вони видавали жахливий звук, схожий на гуркіт грому або рев тисячі левів.

Перед великими святами в лабіринті проводили містерії і приносили ритуальні жертви, в тому числі людські. Так стародавні єгиптяни виявляли свою повагу богу Себеку - величезному крокодилу. У стародавніх манускриптах збереглися відомості про те, що в лабіринті дійсно мешкали крокодили, що досягали 30 метрів в довжину.

За винятком кількох колон, в даний час він повністю зруйнований. Все, що нам про нього відомо, ґрунтується на древніх свідченнях, а також на результатах розкопок, виконаних сером Флиндерсом Петрі, який зробив спробу реконструкції даної споруди.

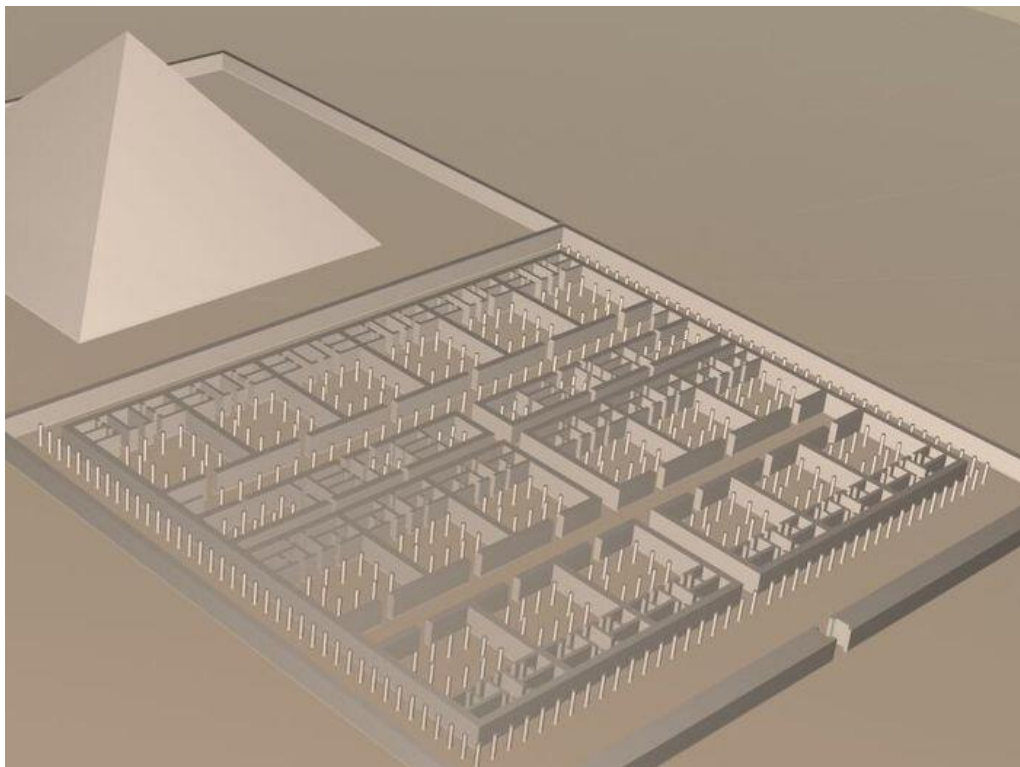


Рис. 4.1. Фаюмский лабіринт



Найперша згадка належить грецькому історичу Геродоту галикарнасском (близько 484- 430 до н. Е.), Він згадує у своїй «Історії», що Єгипет розділений на дванадцять адміністративних округів, якими правлять дванадцять правителів, і далі наводить свої власні враження від цієї споруди (рис. 4.2.):

«І ось вони вирішили залишити спільний пам'ятник, а вирішивши це, спорудили лабіринт трохи вище Мерідово озера, поблизу так званого Міста Крокодилів. Я бачив всередині цей лабіринт: він вище всякого опису. Адже якби зібрати всі стіни і великі споруди, зведені еллінами, то в загальному виявилось б, що на них витрачено менше праці і коштів, ніж на один цей лабіринт. А тим часом храми в Ефесі і на Самосі - вельми чудові. Звичайно, піраміди - це величезні споруди і кожна з них по величині варто багатьох творінь (еллінського будівельного мистецтва), разом узятих, хоча і вони також великі. Однак лабіринт перевершує розмірами і ці піраміди. У ньому двадцять дворів з воротами, розташованими одні проти інших, причому шість звернені на північ, а шість на південь, прилягаючи один до одного. Зовні навколо них проходить одна-єдина стіна. Усередині цієї стіни розташовані покої двох родів: одні підземні, інші над землею, числом 3000, саме по 1500 тих і інших. Надземними покоїв мені самому довелося проходити і оглядати їх, і я говорю про них як очевидець. Про підземні покої знаю лише з розповідей: доглядачі-єгиптяни ні за що не бажали показати мені їх, кажучи, що там знаходяться гробниці царів, воздвигших цей лабіринт, а також гробниці священних крокодилів. Тому-то я кажу про нижніх покоях лише з чуток. Верхні ж покої, які мені довелося бачити, перевершують усі творіння рук людських. Переходи через покої і звивисті проходи через двори, будучи вельми заплутаними, викликають почуття нескінченного подиву: з дворів переходиш в покої, з покоїв в галереї з колонадами, потім знову в покої і звідти знову на подвір'я. Усюди кам'яні даху, так само як і

стіни, а ці стіни покриті безліччю рельєфних зображень. Кожен двір оточений колонами з ретельно прикладанням шматків білого каменю. А на розі в кінці лабіринту споруджена піраміда заввишки 40 оргій, з висіченими на ній величезними фігурами. У піраміду веде підземний хід».

Інформацію про три тисячі приміщень не слід сприймати надто буквально, це скоріше відображення єгипетських вірувань про те, що мандри душі тривають протягом трьох тисяч років. Ці ідеї отримали деяке поширення і в Греції.

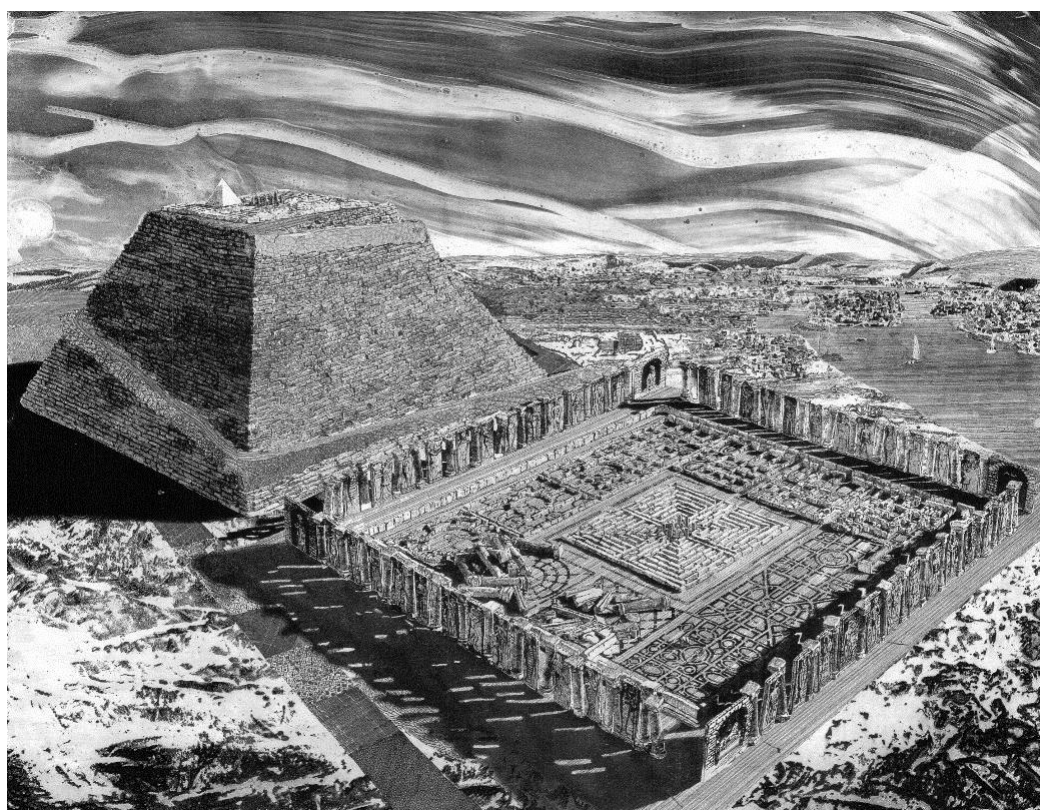


Рис. 4.2. Фаюмський лабіринт з пірамідою

У період між 60 і 57 роками до н. е. грецький історик Діодор Сицилійський тимчасово жив в Єгипті. У своїй «Історичній бібліотеці» світової історії він стверджує, що єгипетський лабіринт знаходиться в

хорошому стані, хоча бачити його на власні очі він ніяк не міг. І якщо критський лабіринт на той час був повністю зруйнований і ніхто з древніх авторів не бачив його.

На відміну від Діодора, грецький географ і історик Страбон Амасейській (близько 64 до н. Е. - 24 н. Е.) Наводить опис, заснований на особистих враженнях. У 25 році до н. е. він в складі пошти префекта Єгипту Гая Корнелія Галла здійснив подорож до Єгипту, про що він детально розповідає в своїй «Географії»:

«Крім того, в цьому номі є лабіринт - споруда, яку можна порівняти з пірамідами, - а поруч з ним гробниця царя, будівельника лабіринту. Близько першої входу в канал, пройшовши вперед 30 або 40 стадій, досягнемо плоскій місцевості в формі трапеції, де знаходиться селище, а також великий палац, що складається з багатьох палацових приміщень, числом стільки, скільки в колишні часи було номів; бо там знаходиться стільки залів, які оточені примикають один до одного колонадами; всі ці колонади розташовані в один ряд і уздовж однієї стіни, яка подібна до довгої стіни з залами перед нею; шляхи, що ведуть до них, знаходяться прямо навпроти стіни. Перед входами в зали знаходиться багато якихось довгих критих склепінь зі звивистими шляхами між ними, так що без провідника жодному іноземцеві не знайти ні входу, ні виходу. Викликає подив, що дах кожного спокою складається з одного каменю і що криті склепіння в ширину точно так же покриті плитами з цільного каменю надзвичайно великий величини, без усякої домішки дерева де-небудь або якого-небудь іншого речовини. Піднявшись на дах невеликої висоти, так як лабіринт одноповерховий, можна бачити кам'яну рівнину, що складається з каменів такий же великий величини; звідси, спустившись знову в зали, можна помітити, що вони розташовані навряд і

спочивають на 27 колонах; стіни їх складені також з каменів не менші за розміром. В кінці цієї будівлі, що займає простір більше стадії, поміщається гробниця - чотирикутна піраміда, кожна сторона якої має близько плефра в ширину при рівній висоті. Ім'я похованого там - Імандес. Кажуть, що така кількість залів побудовано в силу звичаю збиратися тут всім номам відповідно значенням кожного, разом зі своїми жрецькими і жрицями для здійснення жертвоприношень, принесення дарів богам і для судочинства по важливих справах. Кожному ному було відведено призначений йому зал ».

Кілька далі, в 38-му розділі, Страбон призводить опис своєї поїздки до священних крокодилам Арсиної (Крокодилополь). Це місце розташоване поряд з лабіринтом, тому можна припустити, що і лабіринт він теж бачив.

Автори в давнину не мають ніякого єдиного, послідовного визначення цього видатного споруди. Однак оскільки в Єгипті часів фараонів з каменю будувалися тільки святилища і споруди, присвячені культу померлих (гробниці і похоронні храми), - тоді їх всі інші будівлі, включаючи і палаци, будувалися з дерева і глиняних цеглин, - значить, лабіринт ніяк не міг бути палацом, адміністративним центром або пам'ятником (за умови що Геродот, говорячи про «пам'ятнику, монументі», не має на увазі «гробницю, що цілком можливо).

З іншого боку, оскільки фараони XII династії споруджували як гробниць піраміди, єдино можливим призначенням «лабіринту» залишається храм. Згідно вельми правдоподібного пояснення, що приводиться Аланом Б. Ллойдом, він, ймовірно служив і похоронним храмом для Аменемхета III, який був похований в піраміді неподалік, а також храмом, присвяченим деяким богам.

Непереконливим залишається і відповідь на питання про те, як цей «лабіринт» отримав свою назву. Були спроби вивести цей термін з єгипетських слів «al lora-rohun, laperohunt» або «ro-per-ro-henet», що позначають «вхід в храм у озера». Але між цими словами і словом «лабіринт» немає фонетичної відповідності, також і в єгипетських текстах не виявлено нічого схожого.

Висловлювалося також припущення, що тронне ім'я Аменемхета III, Ламарес, елінізований варіант якого звучить як «Лабаріс», походить від назви храму Лабаріса. Таку ймовірність можна виключити, однак це не пояснює суті явища. Більш того, вагомим аргументом проти подібної інтерпретації служить той факт, що Геродот, автор самого раннього письмового джерела, не згадує про Аменемхета III і його тронних іменах. Чи не згадує він і про те, як ця споруда називали самі єгиптяни («Аменемхет живе»). Він просто оповідає про «лабіринті», не вважаючи за потрібне пояснювати, що це таке. Для опису величезного, що вселяє благоговіння, майстерно виконаного кам'яної споруди він використовує грецький термін, немов цей термін висловлює якесь загальне значення, концепцію. Саме такого роду опису наводяться і в усіх інших письмових джерелах, і лише автори пізніших часів згадують про небезпеку заблукати. Отже, можна зробити висновок, що термін «лабіринт» в даному випадку використовується метафорично, він служить назвою для якоїсь споруди, видатного споруди, виконаного з каменю. М. Будимир, вдаючись до історико-лінгвістичної аргументації, прийшов до аналогічного висновку, трактуючи лабіринт як термін, що позначає «будівництво великої величини».

## 4.2. Дослідження схожих відеоігор

Перед тим як починати розробку власних гри, необхідно дослідити інші схожі проекти. Виявити їх позитивні та негативні сторони. Площадкою для дослідження подібних ігор я вибрав Play Market. Це магазин різноманітного софту для телефонів на бази операційної системи Android. Він зручний так як велика кількість ігор там безкоштовна, а також сама платформа більш доступна для розробників. Отже і кількість різних проектів буде вища (на відміну від якості яка може сильно різнитись).

### Maze and More



Рис. 4.3. Інтерфейс гри

Це класична гра про лабіринти. Де режим Classic сподобається як дітям так і дорослим, адже саме їм ця гра нагадає ті самі лабіринти які були у дитячих журналах, і які так весело було проходити олівцем чи ручкою. Осі і тут все доволі просто, необхідно лише свайпнути пальцем в необхідну сторону, і мітка гравця буде автоматично рухатись до наступного перехрестя. Як на мене це дуже вдала система руху для мобільних пристроїв, адже дозволяє грати лише однією рукою.

Мету гри зрозуміти зовсім просто, потрібно лише знайти вихід з лабіринту, важкість котрого кожен раз збільшується. Але це лише у найпростішому – класичному варіанті цієї гри. Розглянемо деякі з них:

- 1) **Enemies** – це режим де гравцеві на протязі гри заважають різні вороги, які постійно переміщуються по лабіринту а не дають спокійно по ньому рухатись. Якщо вони нападуть то рівень потрібно буде приходити по новому. Це вносить певну живість для гри, адже необхідно не тільки знати шлях до виходу, а й не потрапитись ворогу на шляху.

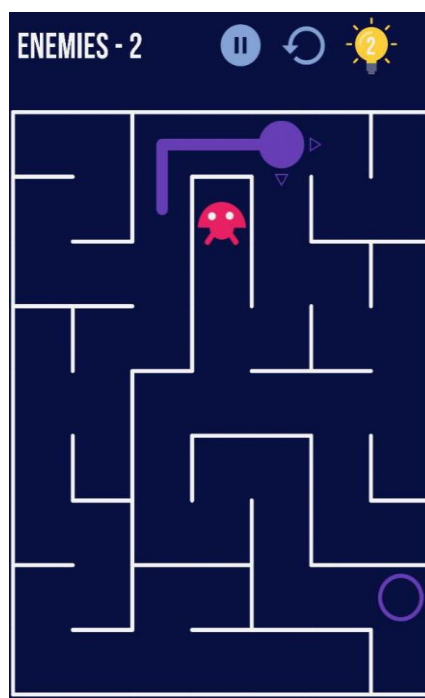


Рис. 4.4. Рівень з ворогом

2) **Darkness** – це режим де гравцеві весь рівень видно лише в самому початку, під час его будування. А надалі лише маленький клаптик рівня біля гравця. Це дійсно дуже цікава особливість, адже навіть маленький рівень буває важко пройти, і це приносить більше задоволення, аніж звичайно проходження. Такий режим дозволяє реалізовувати різні стратегії проходження лабіринту, наприклад завжди притримуючись правої стінки. Рано чи пізно ти вийдеш з нього.

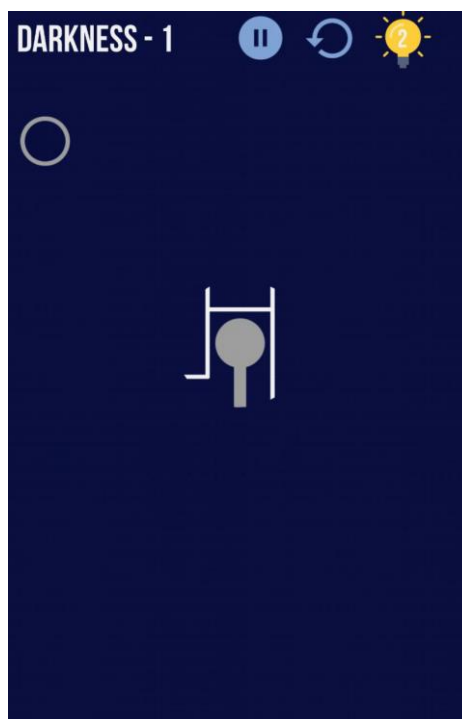


Рис. 4.5. Рівень з темрявою

3) **Traps** – схожий на рівень з ворогом, але тут є точки з яких через певний проміжок вилітають снаряди. Як легко зрозуміти, по аналогії з ворогом якщо їх задіти, то рівень буде необхідно пройти спочатку. В цілому досить цікавий режим, адже необхідно розраховувати час виліту снарядів з різних позицій, і проскочити усі перешкоди.



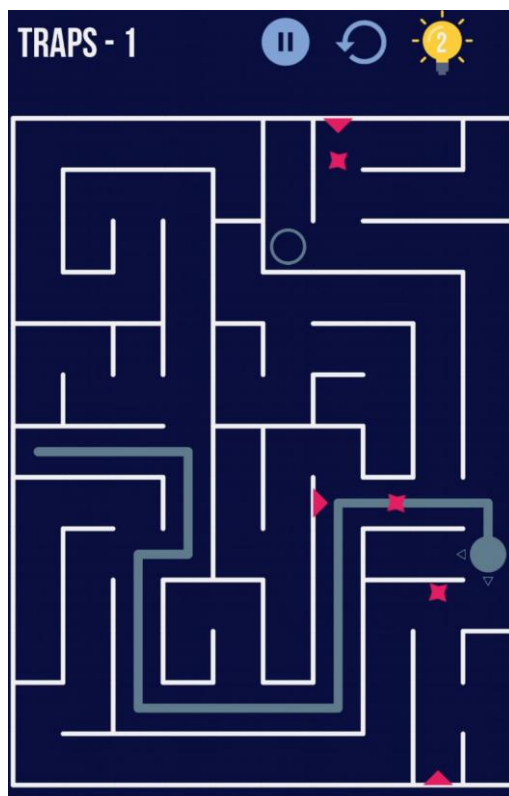


Рис 4.6. Рівень з пастками

В цілому ця гра є дуже гарною реалізацією звичайних лабіринтів. Невигадлива 2D графіка повертає нас в ретро-середу. На гравця чекає велика кількість рівнів із безліччю несподіваних сюрпризів.

Але, для моєї мети, а саме розвитку просторового уявлення, ця гра не дуже підходить адже у кожному режимі (окрім темряви) людина одразу може розрахувати увесь шлях і їй не потрібно запам'ятовувати повороти, перехрестя і т.д. Тобто така реалізація лабіринту не може розвинути в людини просторове мислення. Також у ній відсутній елемент змагань з іншими людьми, лише кількість пройдених рівнів. Цього може бути досить для дорослих, але ніяк не дітей, їм потрібно щось більш наглядне то миттєве.

## Classic Labyrinth

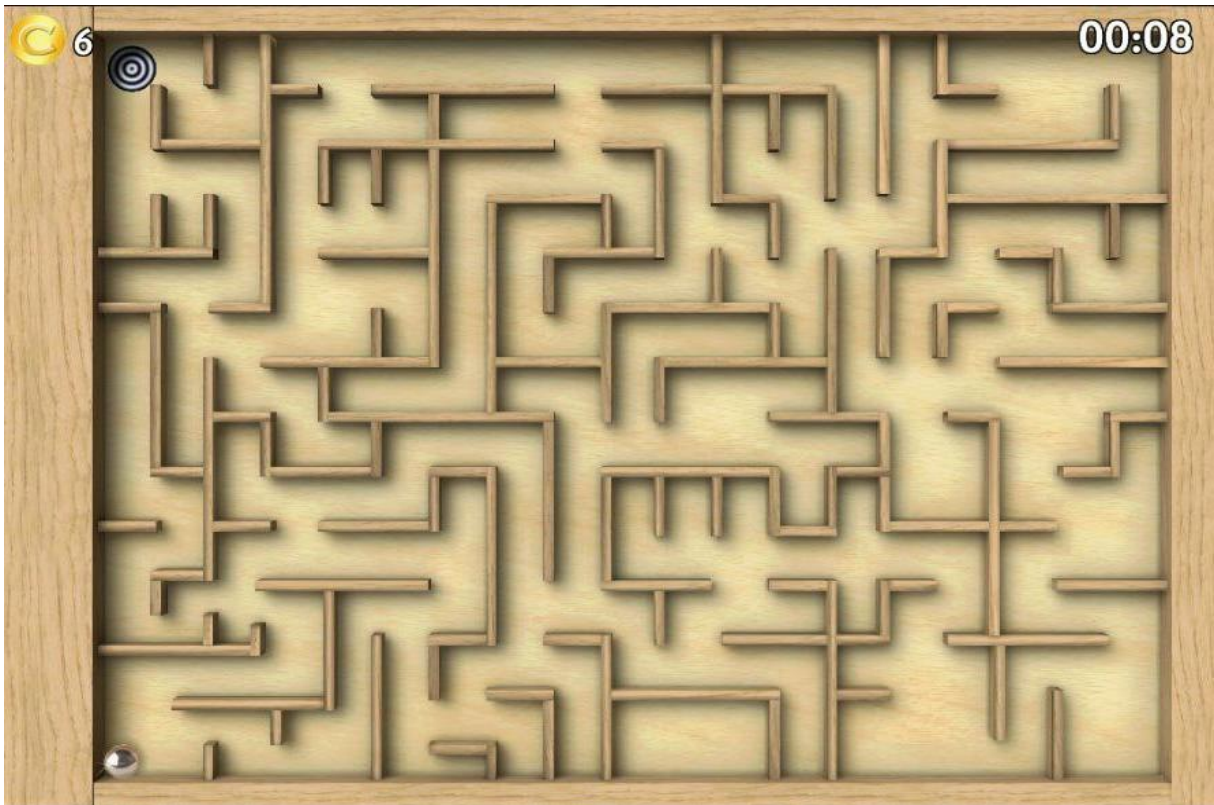


Рис. 4.7. Типовий рівень

Досить популярний різновид ігор з лабіринтом. У даному жанрі частіше за все використовується гіроскоп у телефоні, адже метою гри є закотити м'ячик у дірку виходу. Уникаючи різні перешкоди. На словах звучить досить просто, але насправді зробити це без достатньої координації рухів задача не з простих. На протязі проходження рівнів, постійно зустрічаються різні перешкоди, чи незвична побудова рівню. Найпростіші й одночасно найпопулярніша перешкода.

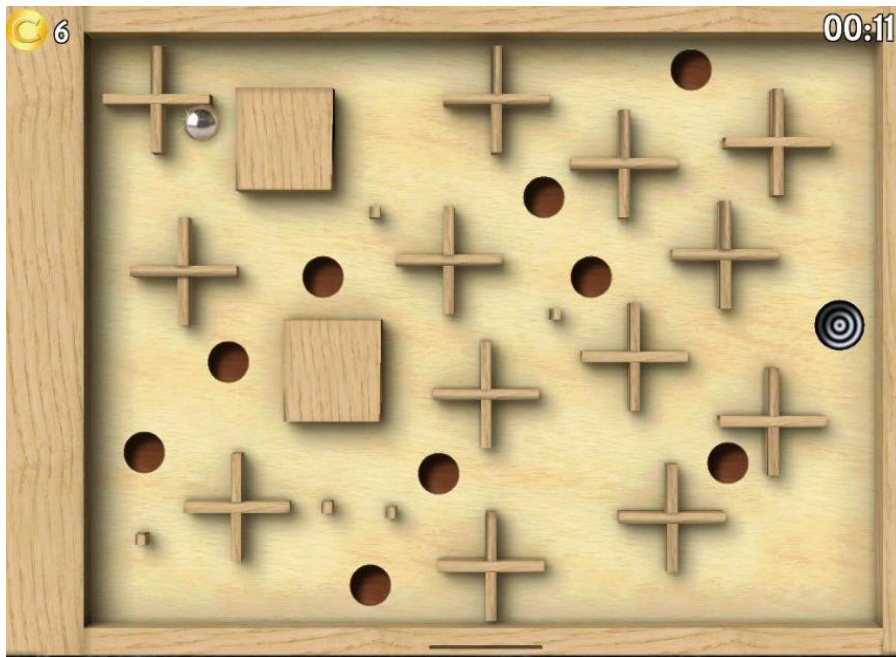


Рис. 4.8. Рівень з дірками-перешкодами

Як видно потрібно дуже акуратно керувати кулею, аби вона не потрапила у западню бо потрібно буде проходити рівень спочатку. Ще одним важким випробуванням є режим темряви. По аналогії з попередньою грою, тут в гравець бачить лише невеличкий шматок поля навколо кулі. Це дійсно складно пройти, адже не у кожного вистачить витримки спокійно та повільно проходити рівні.

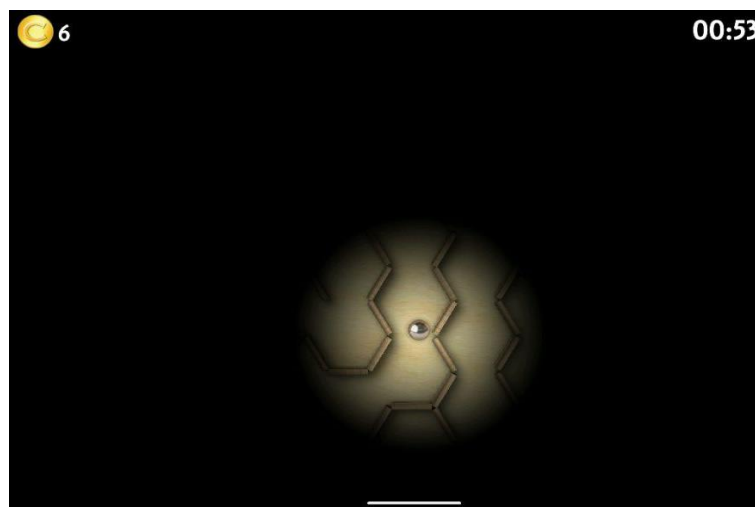


Рис. 4.9. Рівень з темрявою

У цієї гри досить багато переваг:

- реалістична фізика кулі.
- приємна графіка.
- динамічні та цікаві рівні.
- розвиває мілку моторику пальців та швидкість реакції.
- велика кількість рівнів.
- Різноманітність перешкод та режимів.

Вже за це вона заслуговує трохи часу на увагу. Але все ж таки вона не виконує поставлену мною мету, не тренує просторове мислення. Знову ж таки в ній головним завданням в лабіринти не є знайти шлях, а скоріше обійти усі перешкоди найкоротшим шляхом. Це звичайно дуже весело і має свої переваги про які було написано вище, але не підходить для даного дипломного проекту.

### Roll the Ball

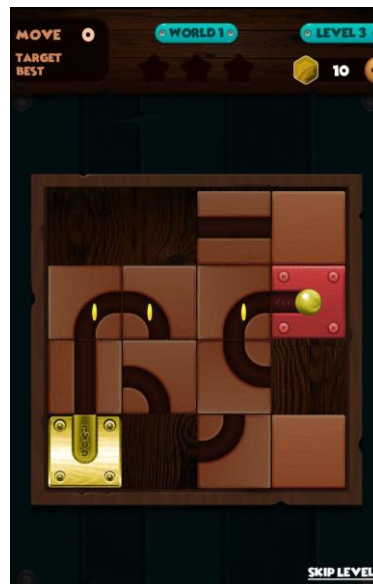


Рис. 4.10. Рівень гри

Такий жанр ігор був популярний ще до створення нормальних відеоігор. Це були пластинки різного розміру, на яких було необхідно пересувати квадратики таким чином, щоб створити шлях по якому зможе прокотитись шарик від старту до фінішу. Чи інший різновид, скласти цифри по порядку, або картинку, суть одну, правильно собі уявляти кінцевий результат, та мати гарне просторове мислення для продумування усіх кроків [2]

Саме дана гра є ідейним наслідником тих старих ігор, але з поправкою на сучасні реалії. Рівнів набагато більше, є деякі нові механіки (буду представлені далі) да и розмір може змінюватись від рівня к рівню.

Ось як буде виглядати зібраний рівень (рис 4.11.) який було видно на рис. 4.10.

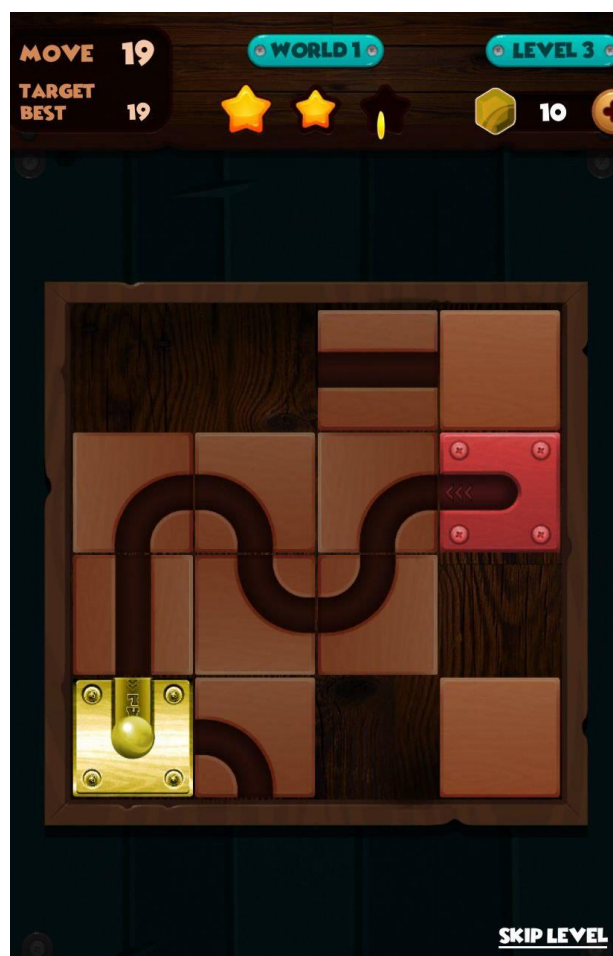


Рис. 4.11. Пройдений рівень

Для ускладнення рівнів було вигадано декілька цікавих механік. Наприклад деякі квадрати неможливо зсунути з місця (рис. 4.12.). Це часто вносить деякі проблеми з пересуванням необхідних елементів, але приносить різноманітність та цікавість, якої була позбавлена стара версія, с одним і тим же рівнем, без новітніх механік.

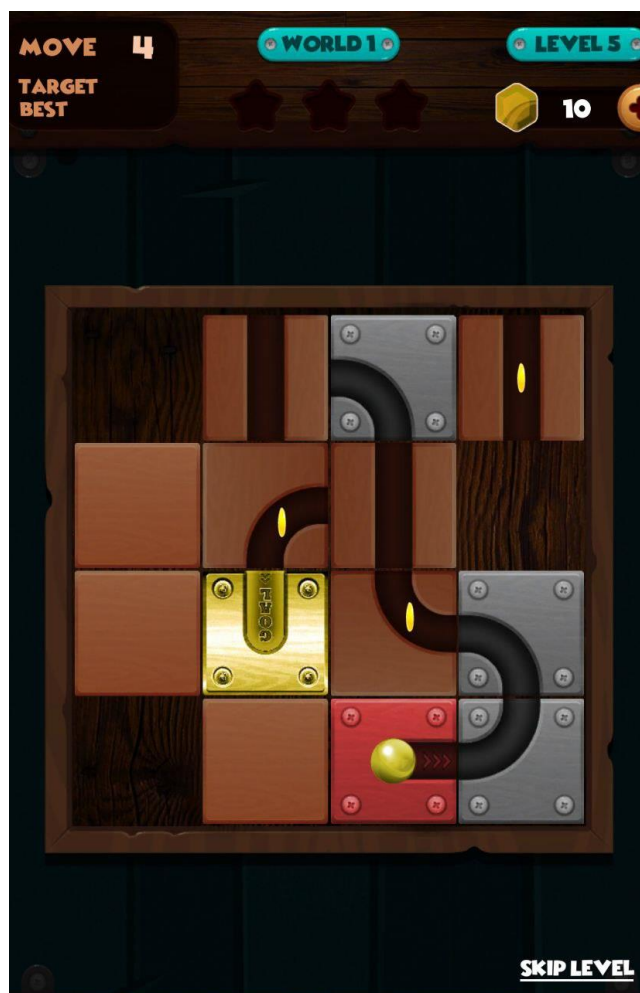


Рис. 4.12. Рівень з нерухомою частиною

Ще одним елементом для розбавлення геймплею, це елементи які можна крутити навколо своєї осі. Тобто тепер необхідно не тільки поставити їх на своє місце, а я правильно повернути.

Даний жанр в цілому дуже гарно реалізований на мобільних пристроях та персональних комп'ютерах. Але він не настільки сильно



залежить від просторового уявлення, яке необхідно для швидкого розвитку в умовах гри за невелику кількість часу. Тому цей тип відеогри також було відхилено.

Отже після перегляду усіх цих ігор, виникають певні умови які повинна виконувати гра на даний час:

- Сучасна 3D графіка
- Різноманітність оточення
- Різні стилі одягу для персонажів
- Мотивація для подальшої гри
- Елемент змагання з іншими
- Нескінченність рівнів, та їх різноманітність
- Правильно модель монетизації

Більшість досліджених ігор в жанрі Лабіринт, мають більш ніж один мінус по даному списку. Найбільш схожими іграми до тієї що буде ідеальною для даного дипломного проекту можна назвати серію ігор про 3D лабіринти: **3D Maze 1,2** (рис.4.12.)



Рис. 4.13. Рівень з гри 3D Maze

Данна гра майже по усім пунктам підходить до висунутих потреб. В неї непогана 3Д графіка, хоча і досить проста як на мене. Досить багато різних рівнів у різних стилістиках. Це додаю більшої цікавості для гравця, адже зміна оточення це завжди приємно у іграх. Осі деякі приклади:



Рис. 4.14. Рівень в пустині



Рис. 4. 15. Льодовий рівень



Спосіб керування досить вдалий:

- джойстик який дозволяє вибирати напрям руху персонажу
- кнопка стрибка яка дозволяє трохи зверху побачити рівень
- кнопка 180° яка дозволяє повернути камеру на 180 градусів

Другим способом керування є свайпи вліво та вправо саме туди автоматично поверне і буде бігти персонаж.

В цілому досить непогана гра, але навіть у ній відсутня велика кількість необхідного. Наприклад:

1. Те що лабіринт можливо розглянути в цілому скоріше мінус аніж плюс. Адже від гравця необхідно лише запам'ятати шлях і пройтись по ньому, тобто просторове мислення дуже слабо використовується при цьому.
2. Графіка все ж таки занадто примітивна, це викликає відчуття недоробленості гри в цілому, або її дешевизни.
3. Не має ніякого різноманіття зовнішнього виду персонажу окрім стандартного.
4. Відсутній елемент змагання з іншими людьми.
5. Велика кількість реклами, що заважає нормально грати.

Усі ці моменти категорично не підходять для мене, але основну концепції 3Д лабіринту можна застосовувати та покращити.

## ВИСНОВОК ДО РОЗДІЛУ 4

Отже для реалізації гри основною концепцією було вибрано лабіринт. Як відомо ще з древніх часів, це дійсно був виклик для можливостей людини, і потребував не аби яких здібностей до орієнтування та просторового мислення. Тому він ідеально підходить до потреб даного проекту.

При розгляданні типових ігор в жанрі лабіринту, було зроблено висновки про їх не вдалість з точки зору необхідних потреб, деякі з них були зроблені дійсно якісно, але то були не ті різновиди лабіринтів що необхідні зараз. Тому було вирішено створювати 3D лабіринт, без повного бачення рівня, з гарною, але не вибагливою графікою. Різноманіттям локацій, та автоматичною генерацією рівня. Це дозволить повисити реіграбельність гри, без докладання великих зусиль ти трати часу.

## РОЗДІЛ 5. Розробка відеогри

У розробці відеогри існує безліч етапів. З ланцюгу не можна виключити жоден з етапів. Адже Кожен важливий по своєму. Найскладніше вже було продумано в попередніх розділах, а саме розробка основної концепції гри. Основні механіки були обрані після аналізу необхідних потреб та вже існуючих відеоігор даного жанру.

Ігровим рушієм для реалізації гри була обрана Unity 3D. Хочу помітити, що не варто власноруч створювати велосипед, адже на просторах інтернету завжди можна знайти різні варіанти реалізації того чи іншого алгоритму. Це дозволить зекономити багато часу на розробці. Також дозволяє розробляти AR і VR ігри.

### 5.1. Необхідне програмне забезпечення

Розпочати роботу з Unity досить легко, необхідно лише завантажити її з офіційного сайту розробника, та встановити його на свій комп'ютер. Адрес сайту досить легко знайти в інтернеті, вибираємо версію а саме Personal. Вона безкоштовна, на відміну від двох інших. Звичайно у них є свою плюси, наприклад:

- Вибір екрану при запуску гри
- Акції на асети у офіційному магазині Asset Store
- Додаткові матеріали, які дозволяють швидше вивчити ігровий рушій
- Покращена система генерації місцевості та флорі

Також одну з платних версій необхідно придбати якщо проект заробляє в рік більше ніж 100 000\$. Це вимоги Unity, для подальшого користування їх продукцією. Це вигідно адже виплати не процентні, а фіксована ціна за комп'ютер.

Кафедра КІТ				НАУ 20 26 27 000 ПЗ			
Виконав	Семенов Н.М.			Розробка відеогри	Літера	аркуш	аркушів
Керівник	Холявкіна Т.В.					91	20
Консульт.					УС 211М 122		
Н. контроль	Райчев І.Е.						

Як вже було сказано, для даного проекту з головою вистачить безкоштовної версії. Вона повністю функціональна, і підходить для створення будь-якої гри, особливо для початківців, та при обмеженому часі.

Зараз для зручного користування Unity, та використання декількох версій одночасно, було створено спеціальний додаток Unity Hub (рис. 5.1.). Це дуже корисний інструмент який дозволяє. Завантажувати нові версії. Запускати потрібні проекти з одного місця, при цьому вибираючи необхідну версії з тих що вже наявна на комп'ютері, або завантажити та встановити іншу, яка буде необхідна. Економія часу, та зручність використання, одразу зробили його необхідною частиною для роботи з Unity.

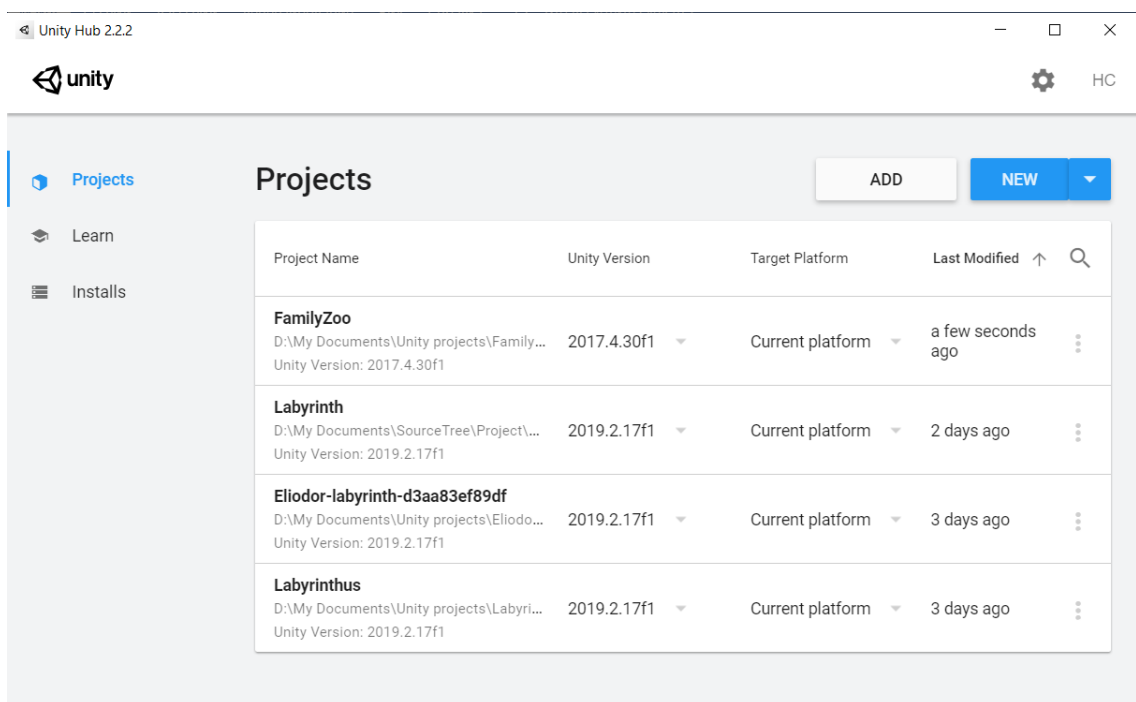


Рис. 5.1. Початковий екран Unity Hub

Зліва я перемикач з трьома основними пунктами меню для роботи с хабом.

- Projects – основне вікно у якому користувач бачить проект. Його

назву. Шлях до нього на комп'ютері. Останню версію Unity з якої він бу запущений. Платформу під яку зараз скомпільований проект. А також останній час запуску проекту (рис. 5.2.).

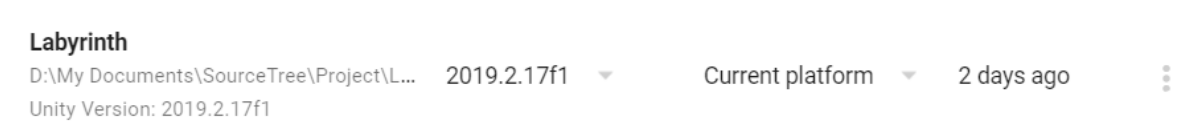


Рис. 5.2. Інформація про проект.

- Learn – частина де у людини є можливість подивитись якісь туторіали, або завантажити вже готовий проект. Це дуже зручно при навчанні, адже набагато зручніше та краще наглядно побачити використання тих чи інших можливостей (рис. 5.3.).

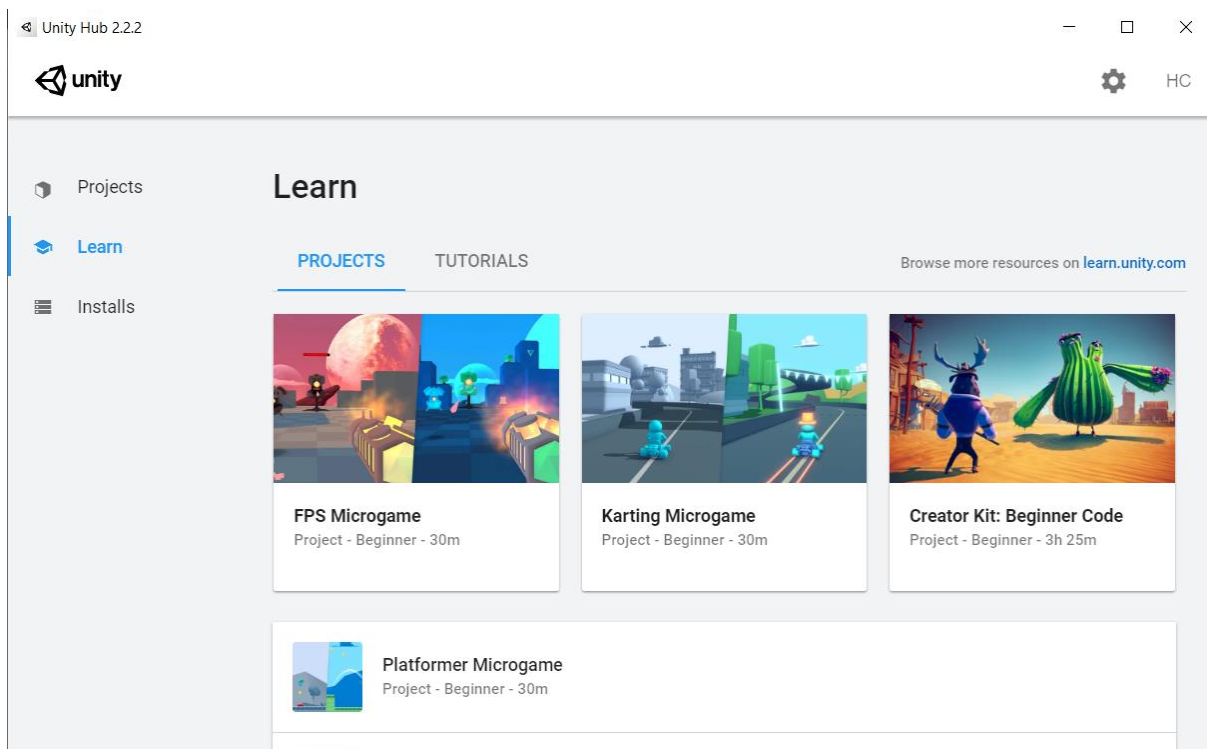


Рис. 5.3. Вікно Learn

- Installs – останнє вікно де можна керувати вже встановленими версіями Unity або встановити нову.

Отже для початку проекту завантажуюємо останню стабільну версію. Раджу звернути увагу на додаткові пункти меню. А саме на можливість одразу завантажити дуже зручну програму для написання коду Visual studio. Це сучасна середовище розробки (IDE) створена Microsoft. Адже основна мова для написання скриптів у Unity це C#. А він був створений майкрософт, тому дана програма досить зручна, і використовується навіть професіоналами. Також рекомендую одразу встановити необхідні компоненти для розробки під Android. Адже це буде друга платформа під яку буде орієнтована гра. Не забуваємо про Android SDK, та Java JDK

Далі натискаємо на клавішу NEW. У новому вікні вибираємо тип проекту, в нашому випадку «3D». Вводимо назву, та місце збереження.

Усе, мінімальний набір для розробки своєї гри встановлено та створено.

## **5.2. Розробка архітектури проекту**

До початку написання основних сценаріїв гри, необхідно продумати основну структуру гри, що повинно бути створено для роботи проекту, хто це буде контролювати, де зберігати інформацію? ПО кожному пункту є безліч варіацій розвитку подій. У кожного з них є як свої плюси так і мінуси. Почнемо з основних моментів та створимо перелік необхідних можливостей.

- Основне меню гри. Чистина відеогри яка відповідає за доступ до різних елементів гри. Можна створити цілу систему яка дозволить у будь який момент часу відкривати нові вікна, поверх інших. Зручно додавати нові не руйнуючі інші зв'язки та майже не торкаючись коду. Але реалізація подібної архітектури займає багато часу. І якщо не передбачається потреби для постійного додавання нового контенту

або елементів, рекомендується значно спростити її. Для більш швидкого завершення проекту.

- Налаштування. Окреме вікно де гравець зможе вибрати певні налаштування гри. Наприклад регуляція звуку. Даний елемент є у кожній грі тому не бачу сенсу довго на ньому затримуватись. Скажу лише що в даному проекті налаштування не є пріоритетною задачею. Тому, можливо, буде відкладено для подальшого покращення.
- Магазин. В нашому випадку він буде вирішувати одразу 2 задачі. Перша це монетизація гри, є гарний спосіб ненав'язливо дати користувачеві можливість ввести певні пошти у гру для свого задоволення. В основному у ньому плануються речі які довозлять змінювати зовнішній облік. Це вирішую другу проблему, а саме відокремлення кожного гравця від іншого. Велика кількість речей та їх комбінація, дозволить зробити майже неповторний стиль для кожного гравця. Також гарним буде створення можливості вибору кольору окремих частин одягу. Це додає ще більшого різноманіття у нарядах гравців.
- Лабіринт. При початку гри, кожен зможе вибрати його рівень складності. Під складністю я розумію розмір лабіринту. Тобто 3x3, 5x5, 7x7 і так далі. Чим вище складність тим більше нагорода, що досить логічно. Кожен наступний рівень складності буде відкриватись при вдалому завершенні наприклад 10 попередніх. Також буде можливість вибрати стиль рівню. Відкриватись вони будуть по такому ж принципу.

Кожен лабіринт має бути згенеровано з використанням рандомайзера, тобто навіть на однаковій складності, шанс того що у людей будуть однакові лабіринти дуже малий. А при збільшенні

складності а отже і розміру, цей шанс і зовсім непомітний. Це вирішить проблему реіграбельності рівнів. Та дозволить зекономити багато часу на розробку індивідуальних рівнів для кожною складності. А також неможливо буде подивитись інструкції в інтернеті з проходженням рівня. І змусить людину дійсно розвивати просторове орієнтування, щоб самому завершити рівень.

Розпочнемо з основного меню (GUI). Зовнішній вигляд було вирішено зробити в стиль досить відомої реалізації лабіринту, яку часто можна зустріти у парках. Лабіринт з кущів, у таких часто люблять грати діти. Тому і задній фон буде з великих кущів. А основні елементи меню ніби вибиті на деревині (рис. 5.4.).



Рис. 5.4. Початкове меню відеогри

Також на ньому буде відображено кількість монет та алмазів які є у гравця. Алмази це додаткова валюта, легше всього яку можна буде просто купити за реальні кошти. А за них вже будуть продаватись деякі ексклюзивні речі для персонажу. Які не можливо буде купити за звичайні



монетки. А також деякі інші товари.

Кожна за написів інтерактивна і відкриє необхідне вікно для користувача. Реалізувати це у Unity досить просто. Кожне віно, це окремий об'єкт який існують у 2д просторі гри. Як і будь-який елемент його можна відключити або навпаки включити. А саму кнопку також легко створити з контекстного меню (рис. 5.5.). Натискаємо правою клавішею мишки на батьківській елемент у вікні Hierarchy та вибираємо необхідні пункти меню.

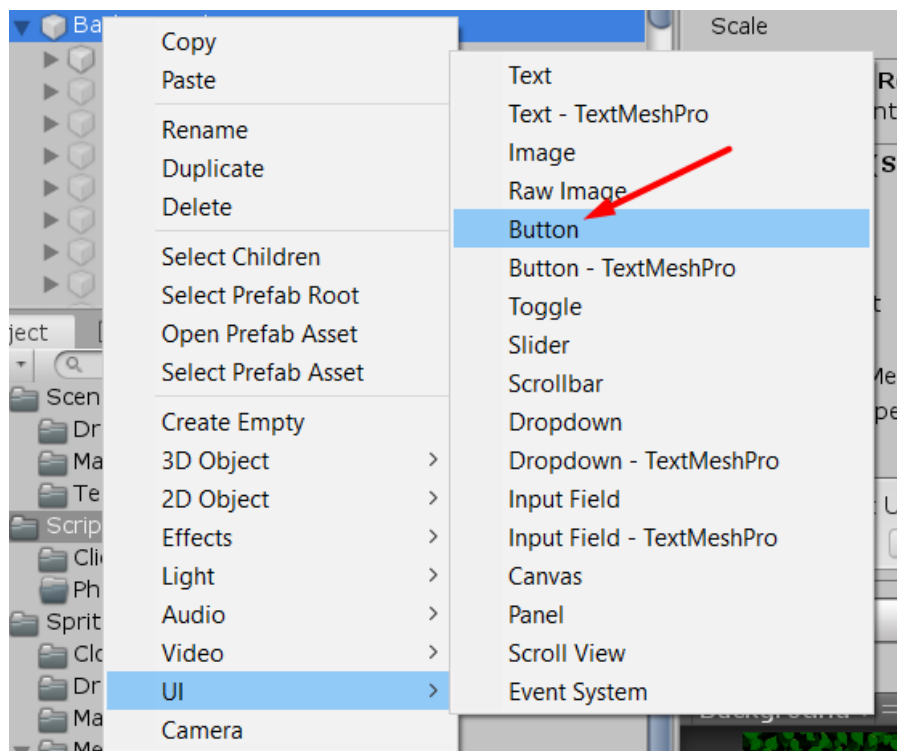


Рис. 5.5. Створення кнопки

Далі налаштуємо зовнішній вигляд кнопки, вибираючи тип анімації, колір при різних діях, а також додаємо текст. За нього відповідає окремий елемент від асету TextMeshPro. Це один з найпотужніших компонентів для роботи з текстом. Його можливості дійсно неймовірні. Майже все ще захоче зробити розробник ігор, можливо реалізувати завдяки

цьому компоненту. Осі і я завдяки ньому виберу потрібний стиль, та колір, з мінімальною окантовкою літер.

Для того щоб при натиску на кнопку відбувалась потрібна нам дія, в нашому випадку ввімкнення певного вікна. Необхідно в полі OnClick (рис. 5.6.) натиснути на плюсики, та перетягнути це вікно (яке буде вмикатись) у поле, вибравши необхідні налаштування.

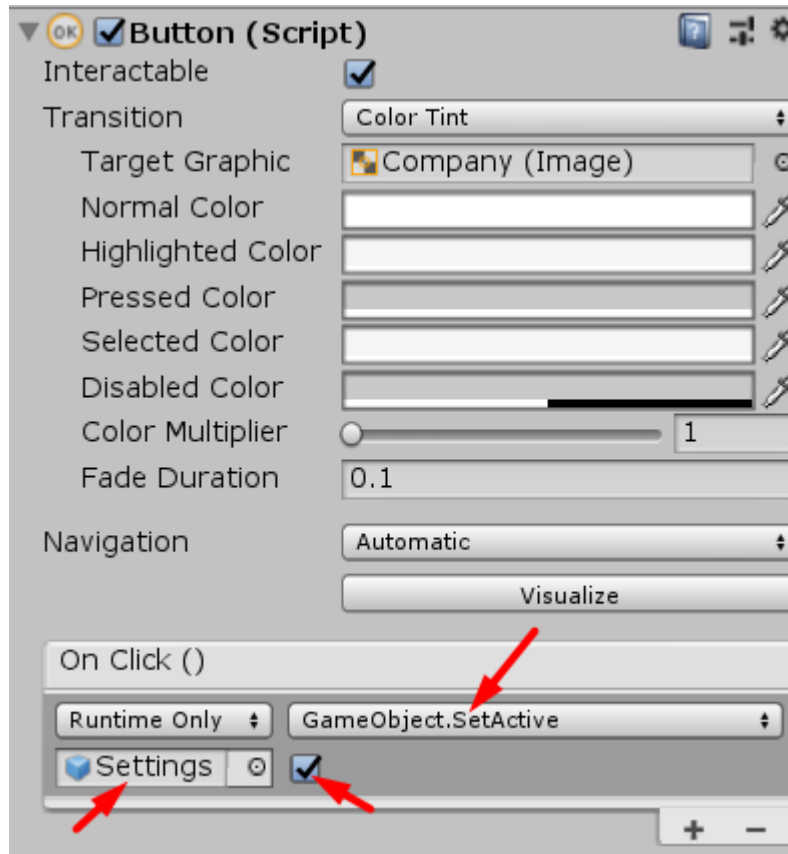


Рис. 5.6. Налаштування кнопки

Вибираємо пункт `GameObject.SetActive`, та активуємо галочку. Це означає що при кліку на дану кнопку, він буде активувати необхідний нам об'єкт. Тобто це саме те що нам було необхідно виконати. Осі так, за пару рухів, ми реалізували частину меню гри, зовсім без коду та зайвих рухів. Саме за це сучасні розробники люблять ігрові рушії.

Основним сценарієм для роботи ігрового меню буде `Menu`. Усі

скрипти (сценарії) зберігаємо за адресою Labyrinth -> Scripts. Так буде зручно орієнтуватися у структурі папок проекту. У цьому скрипті будуть методи які відповідають за:

- Відкриття та закриття деяких панелей
- Оновлення тексту з показниками валют які має гравець
- Отримання розміру останнього вибраного розміру та стилю лабіринту
- Вимкнення гри
- Збереження вибраного одягу
- Додавання вибраної валюти
- Пауза у грі
- Відкриття рюкзаку у грі
- Встановлення марки та стрілки (підказки)

Це один з найрізноманітніших скриптів, адже відповідає за багато різних речей.

### *Контролер звуку*

Це буде скрипт який відповідає за музику та звуки у всій грі. Він відповідає за перемикання музики між сценами. Та за ввімкнення та вимкнення звуків або музики в цілому у грі.

Для зручного доступу до його можливостей, на усіх схожих сценаріях буде використовуватись дуже відомий паттерн програмування «Одинак» або «Singleton». Він, відноситься до класу твірних шаблонів. Гарантує, що клас матиме тільки один екземпляр, і забезпечує глобальну точку доступу до цього екземпляра.

Слід використовувати шаблон Одинак коли:

- повинен бути тільки один екземпляр деякого класу, що легко доступний всім клієнтам;
- єдиний екземпляр повинен розширюватись шляхом успадкування, та клієнтам потрібно мати можливість працювати з розширеним екземпляром не змінюючи свій код.

Цей паттерн дозволить нам звертатись до цього класу з будь якого місця в кодї. Це дійсно дуже зручно при розробці та використанні одного контролеру для окремих елементів гри. Для керування звуками із гри було створено окрему вікно з необхідними кнопками. Його позначення досить типове для налаштувань, и виглядає як шестерня. При натисканні відкривається необхідне вікно (рис.5.7.). Далі натискаючи на кнопки можна вмикати чи вимикати звук, також теперішній стан налаштувань можна буде зрозуміти по картинці на кнопці.

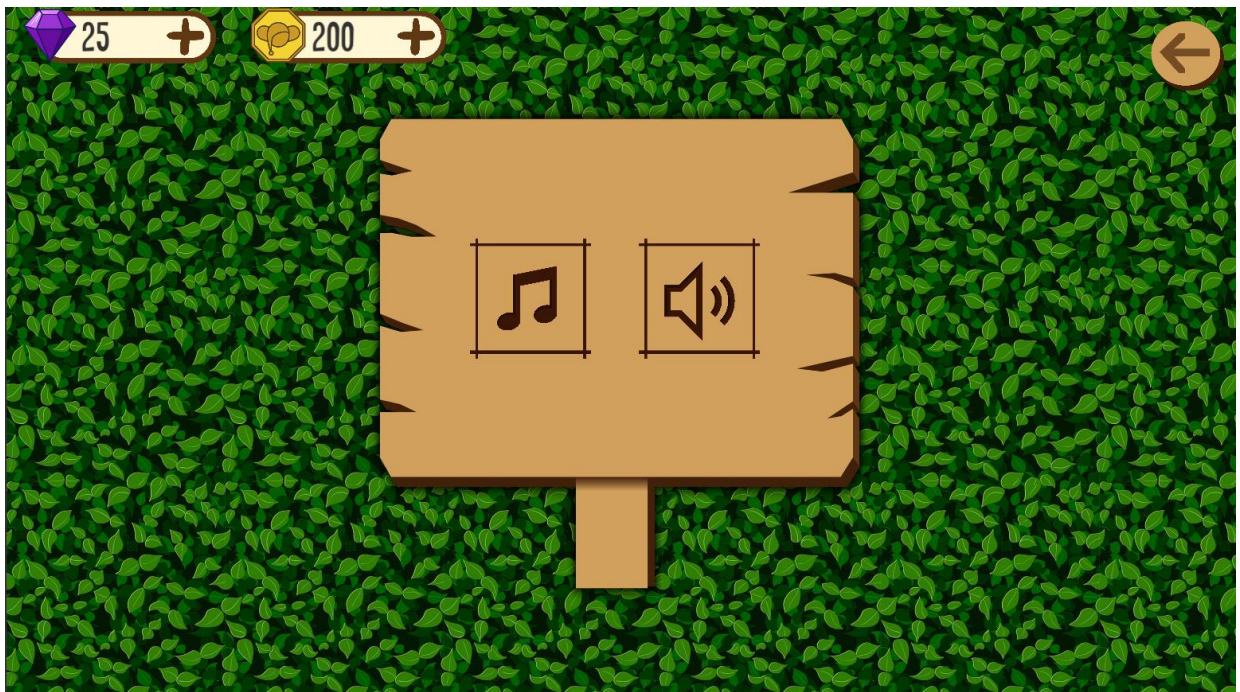


Рис. 5.7. Вікно налаштувань

## *База даних*

Способів збереження даних існую велика кількість. Можна застосувати усім відомі бази даних, які досить непогано працюють із Unity. Можна створити текстовий файл з власною структурою зберігання інформації у форматі JSON. Це один з найпопулярніших текстових форматів для обміну даними. Его суть полягає в тому що є будь який клас або структура даних, яку можна перевести у текст та зберігати як зручно а потім зчитати и отримати потрібну інформацію. Він зручний при відносно не великій кількості даних, адже зчитуватись будуть усі дані одночасно, й при великому об'єму будуть займати багато місця в пам'яті [1].

Також, частково, можна зберігати деяку інформацію у самій юніті, використовуючи якийсь існуючий об'єкт з певним скриптом. Наприклад в ньому будуть зберігатись посилання на часто використовувані об'єкти, та публічні змінні які необхідно, або зручно змінювати у самій Unity, без редагування додаткових файлів.

Для зберігання невеликої кількості даних, можна використовувати внутрішню розробку Юніті під назвою PlayerPrefs. Він дозволяє зберігати прості типи даних, такі як: int, float, string на різних платформах по різному. Наприклад на Windows для цього використовується реєстр. Це досить зручно адже можна у текстовому форматі зберігати дані у форматі JSON, і легко керувати ними. Саме такий спосіб і було обрано. Для цього було створено сценарій Database (рис.5.8.). Який для кожної сцени буде зберігати необхідні для її роботи посилання та компоненти. А також буде містити поля для редагування змінних які будуть нам необхідні.

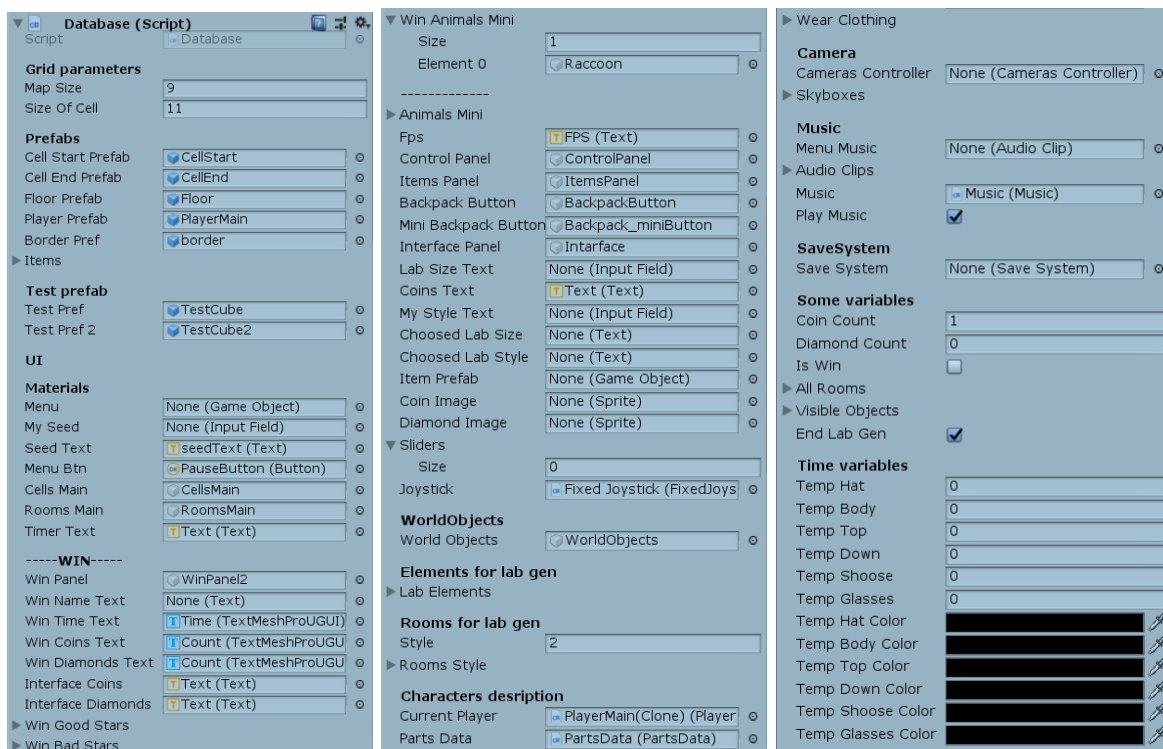


Рис. 5.8. Database

### 5.3. Створення лабіринту

Для початку треба вирішити який алгоритм побудови лабіринтів буде використано. Розглянемо декілька з них щоб обрати підходящий.

#### *Алгоритм двійкового дерева*

Його суть полягає у тому, щоб прокласти шлях у випадковому напрямку з кожної клітини поля: в класичній реалізації або нагору, або вправо (залежить від обраного зміщення). Ми обробляємо тільки 1 клітину за одиницю часу, отже, ми можемо генерувати лабіринти нескінченного розміру, зберігаючи лише кінцевий результат (лабіринт) без необхідності зберігати будь-яку побічну інформацію. Такий спосіб генерації має два побічних ефекту:

1. Лабіринти володіють сильним діагональним зміщенням і відсутністю тупиків в його напрямку. Подивіться на скріншоти вище і Ви побачите, що

кожен з коридорів прагне до правої верхньої клітці, і, як підсумок, має рівно один шлях до неї, і ніде на шляху немає тупика:

2. Два порожніх коридору по сторонам лабіринту. Коли алгоритм «прокопується» до кінця рядка / стовпця, йому не залишається вибору, окрім як продовжити шлях в одному єдиному напрямку, створюючи порожні «кордону».

До слова, назва не просто так збігається зі структурою даних. Результат його роботи - випадкове двійкове дерево, в якому з кожної клітини (вершини) є рівно 1 шлях у напрямку до кореня (батьківської вершини), і, відповідно, рівно 1 шлях до будь-якої іншої клітці. Як наслідок, будь-яка клітина має не більше 3 з'єднань зі своїми сусідами.

Даний алгоритм не підходить нам адже має прямі коридори, ті не дуже гарну структуру самого лабіринту.

### *Алгоритм «Sidewinder»*

Алгоритм з неперекладним назвою Sidewinder по своїй роботі дуже схожий на алгоритм двійкового дерева, в тому відміню, що в ньому немає характерного зміщення по діагоналі, одного порожнього коридору і клітини ми розглядаємо не окремо, а множинами. Лабіринти виходять з переважно вертикальним або горизонтальним зміщенням (в залежності від реалізації), з відсутністю тупиків в їхньому напрямку. У порівнянні зі своїм більш примітивним побратимом, зсув не так помітно і більше схоже на «спіраль», яка плавно змінює вертикальні і горизонтальні коридори.

Що стосується побічних ефектів, то Sidewinder створює тільки один порожній коридор на одному боці, замість двох. Починаючи створення множин з першого ряду поля, у нас відсутня можливість прокопати шлях наверх, так як ми знаходимося в самому крайньому вертикальному

положенні і спроба піти вище призведе до виходу за межі поля. Але і якщо ми будемо організовувати безлічі без виходу по вертикалі, ми створимо кілька ізольованих один від одного областей.

Це більш досконалий алгоритм, але для нас також не підходить, адже прямих шляхів без одиноких кімнат майже немає.

#### «Ідеальний» лабіринт

Це алгоритм, заснований на бектрекінгу, що дозволяє створювати лабіринти без циклів, що мають єдиний шлях між двома точками. Алгоритм не найшвидший, досить вимогливий до ресурсів, у порівнянні з [алгоритмом Ейлера](#) або Круськала, але дуже простий у реалізації і дозволяє створювати гіллясті лабіринти з дуже довгими тупиковими відгалуженнями.

#### Опис алгоритму

Зауваження: передбачається, що спочатку у кожній клітині є стінки з усіх чотирьох сторін, які відокремлюють її від сусідніх клітин.

1. Зробіть початкову клітку поточної і відзначте її як відвідану.
2. Поки є невідвідані клітини
  1. Якщо поточна клітина має невідвіданих «сусідів»
    1. Проштовхніть поточну клітку в стек
    2. Виберіть випадкову клітку з сусідніх
    3. Приберіть стінку між поточною кліткою і обраної
    4. Зробіть обрану клітку поточної і відзначте її як відвідану .
  2. Інакше якщо стік не порожній
    1. Висмикніть клітку з стека
    2. Зробіть її поточної
  3. Інакше



1. Оберіть випадкову невідвідану клітку, зробіть її поточною і відзначте як відвідану.

Ви, мабуть, помітили що при виконанні умови 3, готовий лабіринт найімовірніше матиме ізольовану область.

Ця умова включено в алгоритм як виняток, на практиці при нормальній роботі алгоритму і правильних вихідних даних, воно не виконується ніколи.

### Ілюстрація роботи

1) Створюємо початкову матрицю (рис.5.9.):

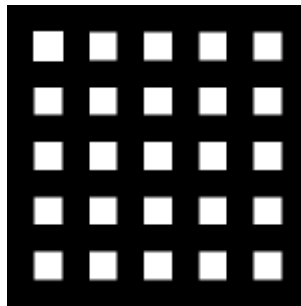


Рис.5.9. Початкова матриця

2) Вибираємо початкову точку стартовою (рис. 5.10.):

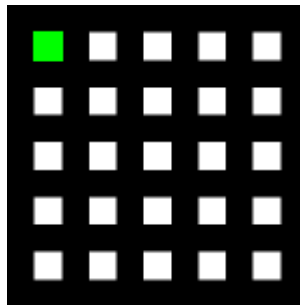


Рис. 5.10. Початкова точка

3) Переміщаємося до випадкового невідвіданого сусіда, поки такі є (рис. 5.11.):

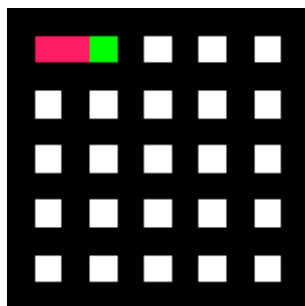


Рис. 5.11. Шукаємо сусідів

- 4) Коли невідвідані сусіди закінчились на основному шляху, повертаймося до будь-якої з попередніх клітинок у якої є такі сусіди (рис. 5.12)

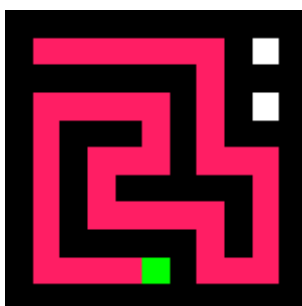


Рис. 5.12. Пошук невідведаних сусідів

- 5) Невідвіданий сусід є, переміщаємося до нього (рис. 5.13.):

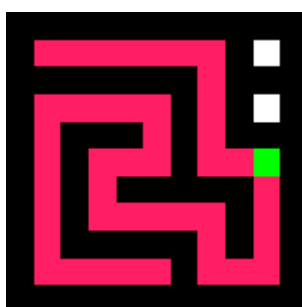


Рис. 5.13. Невідвідана клітинка

- 6) Не відвіданих клітинок більше немає. Лабіринт згенеровано (рис. 5.14.)

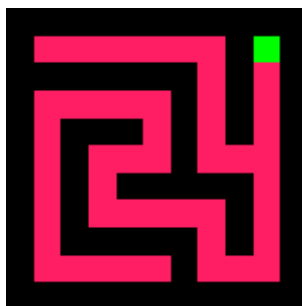


Рис. 5.14. Згенерований лабіринт

Уривок кода, який відповідає за генерацію:

```
cell startCell = { 1, 1 }
cell currentCell = startCell;
cell neighbourCell;
do{ cellString Neighbours = getNeighbours(width, height, maze, startPoint, 2);
if(Neighbours.size != 0)
{ // якщо у клітинки є невідвідані сусіди
randNum = randomRange(0, Neighbours.size-1);
neighbourCell = cellStringNeighbours.cells[randNum]; //вибираємо
випадкового сусіда
push(d.startPoint); //зберігаємо клітинку
maze = removeWall(currentCell, neighbourCell, maze); //Прибираємо стінку
між нинішньою та наступною клітинкою
currentCell = neighbourCell; //робимо сусіда поточною клітиною, її
відмічаємо як відвідану
maze = setMode(d.startPoint, d.maze, VISITED);
free(cellStringNeighbours.cells);
}
else if(stackSize > 0)
{ //якщо немає сусідів повертаємося до попередньої клітинки
startPoint = pop();
```

```

}
else{ //вибираємо одну з не відмічених
cellString cellStringUnvisited = getUnvisitedCells(width, height, maze);
randNum = randomRange(0, cellStringUnvisited.size-1);
currentCell = cellStringUnvisited.cells[randNum];
free(cellStringUnvisited.cells);
}
while(unvisitedCount() > 0);

```

Як видно, реалізація алгоритму проста і абстрактна від теорії, як то кажуть, «впорається навіть дитина».

Для генерації різноманітних лабіринтів, але з можливістю повторити їх при потребі, використовується внутрішній генератор випадкових чисел на основі так званого насіння. Seed це певне число, найчастіше в межах звичайного int, яке подається в генератор випадкових чисел на початку його роботи. І алгоритм працює таким чином, що при однаковому насінні, та кількості запитів, ми постійно будемо отримувати одні й ті самі цифри. Тобто це випадкові значення, але при необхідності їх можна повторити. Це дуже зручно при налаштуванні лабіринту. Щоб не генерувати його постійно по новому для знайдення необхідної ситуації, а постійно генерувати вже відомий варіант. Це дозволяє дуже чітко спостерігати за змінами лабіринту, при правці коду.

Далі додаємо необхідні модельки можливих кімнат, які були зроблені у Blender, формат збереження моделей - FBX. Для нормальної роботи лабіринту необхідно 5 видів клітинок.

У формі літер I, L, U, X, Y. Завдяки цим 5 вдам кімнат, можна створити найрізноманітніші лабіринти, будь-якого розміру. Зробивши декілька

варіантів текстур та схожих моделей, отримуємо декілька стилів лабіринту.

Далі доаємо необхідний інтерфейс користувача. На мобільних пристроях це буде джойстик, а на комп'ютері клавіші WASD. А також додамо декілька допоміжних засобів для гравця, а саме: крестик, щоб помічати вже відвідані участки мапи, та стрілка яка покаже гравцеві де саме буде вихід з лабіринту. Приклад стилів з інтерфейсом на рис. 5.15.

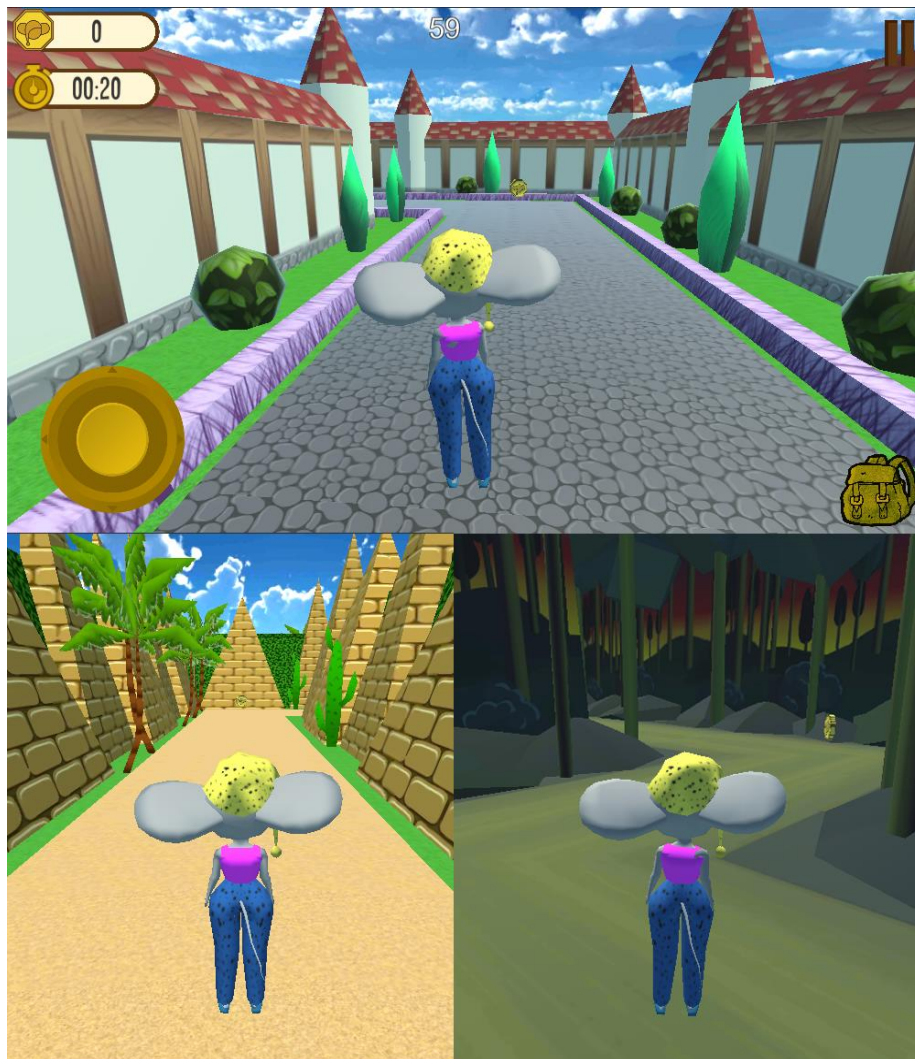


Рис. 5.15. Приклад зовнішнього виду гри

Останнім кроком у створенні гри, є створення магазину (рис. 5.16.) де можливо буде вибирати зовнішній вигляд персонажу та купувати новий одяг. У базовому представлення одяг це також 3D моделі, які є універсальними для певного скелету. Було вирішено зробити 5 видів одягу,

та можливість змінювати сам скелет, з мишки на іншого персонажу. ItemsCreator це сценарій який створює усі доступні речі. До тих що не куплені, додається ціна з вказаною валютою. Натепер речей не так багато, але асортимент планується розширити в майбутньому.

Також було реалізовано можливість вибору кольору для кожної речі, що також додає різноманітності кожному персонажу.



Рис. 5.16. Магазин у грі

Компілювання гри від різні платформи дуже просте, необхідно лише вибрати необхідну платформу у меню Build Settings, та натиснути Build. Можливі варіанти з додатковими кроками, але Unity усе продемонструє.

## **ВИСНОВОК ДО РОЗДІЛУ 5**

Цей розділ гарно демонструє як ігровий рушій може спростити розробку відеогри. Реалізація графічного інтерфейсу користувача, майже повністю можлива без зайвого коду. Додавання моделей легко робити з будь якого редактору. Загалом реалізація лабіринту вважаю досить якісною. Як з точки зору коду, так і візуальної частини. Усі ці компоненти впливають на якість гри, яку отримуємо у фіналі розробки. Не варто нехтувати будь яким з цих компонентів. У моєму випадку я намагався передбачити велику кількість необхідних речей, але не кожному з них встиг реалізувати. Тому в цього проекту є перспективи для розвитку.

## ВИСНОВКИ

У цьому дипломному проекті було створено відеогру, яка повинна допомогти людям розвивати просторове мислення. Завдяки великій підготовчій роботі, та прочитанню великої кількості теоретичних матеріалів точно встановлено, що розвиток людини завдяки грі, має реальні шанси впоратись із завданням.

У сучасних реаліях саме відеогра для декількох найпопулярніших платформ буде мати шанси як стати відомою так і допомогти людям. Монетизація проекту потрібна для збору коштів на подальший розвиток проекту. Наприклад для якісної підтримки онлайн гри, необхідно орендувати потужні сервери, а це гроші які потрібно сплачувати кожного місяця. Дана гра має великий простір для подальшого вдосконалення та розвитку. Як вже сказано необхідно додати можливість грати онлайн, отримувати внутрішню валюту за перегляд реклами. Додати історію для персонажу і світу в цілому.

Дана гра розрахована як на дітей так і на дорослих, зараз рамки жанру та стилістики не такі обмежені у голові дорослих, щоб сприймати мультяшну графіку як щось дитяче. Адже у будь якому стилі можна передати певний смисл.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://habr.com/ru/post/262345/>
2. <https://habr.com/post/246605/>
3. [https://pikabu.ru/story/fayumskiy\\_labirint\\_7043563](https://pikabu.ru/story/fayumskiy_labirint_7043563)
4. <http://develop-unity3d.blogspot.com/2016/01/Silnie-storoni-Unity.html>
5. <https://itproger.com/course/unity-csharp>
6. <https://habr.com/ru/post/112287/>
7. <https://www.the-village.ru/village/city/at-a-glance/144677-prilozheniya-dlya-razvitiya-voobrazheniya>
8. <https://www.mosigra.ru/page/igry-na-prostranstvennoe-myshlenie/>
9. <https://childdevelop.com.ua/articles/develop/1759/>
10. [https://knowledge.allbest.ru/pedagogics/3c0a65635a3bc69b4d43b89521316d37\\_0.html](https://knowledge.allbest.ru/pedagogics/3c0a65635a3bc69b4d43b89521316d37_0.html)
11. <https://logiclike.com/math-logic/interesno-polezno/doshkolnik-prostranstvennoe-myshlenie>

## Menu.cs

```

using System.Collections;
using System.Collections.Generic;
using System;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using TMPro;

public class Menu : MonoBehaviour
{
    private SaveInforms saveInforms;
    public int seed;
    public ChangeSprite musicButton;
    public ChangeSprite soundButton;

    private static Menu instance;
    public static Menu Instance { get {
return instance ? instance : instance =
FindObjectOfType<Menu>(); } }

    void Awake()
    {
        instance = instance ? instance
: this;
    }

    private void Start()
    {
        saveInforms =
SaveSystem.Instance.Saves;
        if
(Database.Instance.labSizeText != null)
Database.Instance.labSizeText.text =
saveInforms.choosedLabyrinthSize.ToStri
ng();
        if
(Database.Instance.myStyleText != null)
Database.Instance.myStyleText.text =
saveInforms.choosedLabyrinthStyle.ToStr
ing();
        if
(Database.Instance.interfaceCoins !=
null &&
Database.Instance.interfaceDiamonds !=
null)
        {
            UpdateMoneyText();
        }

        if(musicButton != null)
musicButton.SetSprite(!SoundManager.Ins
tance.musicSource.mute);
    }

    if(soundButton != null)
soundButton.SetSprite(!SoundManager.Ins
tance.soundSource.mute);
    }

    public void UpdateMoneyText()
    {
        Database.Instance.interfaceCoins.text =
SaveSystem.Instance.Saves.coinsAmount.T
oString();

        Database.Instance.interfaceDiamonds.tex
t =
SaveSystem.Instance.Saves.diamondsAmoun
t.ToString();
    }

    public void RandomButton(GameObject
sceneLoading)
    {
        System.Random rnd = new
System.Random();
        seed = rnd.Next(0, 999999999);

        SaveSystem.Instance.SaveSeed(seed);
        sceneLoading.SetActive(true);
    }

    public void CloseMenuPanel()
    {
        Database.Instance.menuBtn.gameObject.Se
tActive(true);

        Database.Instance.menu.SetActive(false)
;
    }

    public void OpenMenu()
    {
        Database.Instance.menuBtn.gameObject.Se
tActive(false);

        Database.Instance.menu.SetActive(true);
        Database.Instance.mySeed.text =
seed.ToString();
    }

    public void SetMark()
    {
        GameHelper.Instance.InstantiateMark();
    }
}

```

```

    }

    public void GetWay()
    {

GameHelper.Instance.InstantiateArrow();
    }

    public void OpenBackpack()
    {

Database.Instance.itemsPanel.SetActive(
true);

Database.Instance.backpackButton.SetAct
ive(false);

Database.Instance.miniBackpackButton.Se
tActive(true);
    }

    public void CloseBackpack()
    {

Database.Instance.itemsPanel.SetActive(
false);

Database.Instance.backpackButton.SetAct
ive(true);

Database.Instance.miniBackpackButton.Se
tActive(false);
    }

    public void SaveLabSize(int size)
    {

SaveSystem.Instance.SaveLvlSize(size);
    }

    public void SaveLabStyle(int style)
    {

SaveSystem.Instance.SaveLvlStyle(style)
;
    }

    public void ClosePanel(GameObject
panel)
    {
        panel.SetActive(false);
    }

    public void OpenPanel(GameObject
panel)
    {
        panel.SetActive(true);
    }

    public void GetParameters()

```

```

    {
        GetMySize();
        GetMyStyle();
    }

    public void GetMySize()
    {
        string size = "";
        if
(SaveSystem.Instance.Saves.choosedLabyr
inthSize == 0) size = "5";
        else if
(SaveSystem.Instance.Saves.choosedLabyr
inthSize == 1) size = "9";
        else if
(SaveSystem.Instance.Saves.choosedLabyr
inthSize == 2) size = "13";
        else if
(SaveSystem.Instance.Saves.choosedLabyr
inthSize == 3) size = "17";
        else if
(SaveSystem.Instance.Saves.choosedLabyr
inthSize == 4) size = "21";

Database.Instance.choosedLabSize.text =
"Your Size: " + size;
    }

    public void GetMyStyle()
    {
        string style = "";
        if
(SaveSystem.Instance.Saves.choosedLabyr
inthStyle == 0)
        {
            style = "Provance";
        }
        else if
(SaveSystem.Instance.Saves.choosedLabyr
inthStyle == 1)
        {
            style = "Egypt";
        }

Database.Instance.choosedLabStyle.text
= "Your Style: " + style;
    }

    public void BackToMenu()
    {
        SceneManager.LoadScene(0);
    }

    public void ExitGame()
    {
        Application.Quit();
    }

    public void OpenScene(int num)

```

```

    {
        SceneManager.LoadScene(num);
    }

    public void SaveChooosedClothes()
    {
        SaveSystem.Instance.ChooseClothes(0,
        Database.Instance.tempBody);

        SaveSystem.Instance.ChooseClothes(1,
        Database.Instance.tempHat);

        SaveSystem.Instance.ChooseClothes(2,
        Database.Instance.tempTop);

        SaveSystem.Instance.ChooseClothes(3,
        Database.Instance.tempDown);

        SaveSystem.Instance.ChooseClothes(4,
        Database.Instance.tempShoose);

        SaveSystem.Instance.ChooseClothes(5,
        Database.Instance.tempGlasses);
    }

    public void AddCoins(int coins)
    {
        SaveSystem.Instance.SaveCoins(SaveSystem.Instance.Saves.coinsAmount + coins);
        UpdateMoneyText();
    }

    public void AddDiamonds(int diamonds)
    {
        SaveSystem.Instance.SaveDiamonds(SaveSystem.Instance.Saves.diamondsAmount + diamonds);
        UpdateMoneyText();
    }

    public void PauseGame(bool pause)
    {
        GameHelper.Instance.isPause = pause;
    }

    #region Win Menu

    public void WinButton()
    {
        SaveSystem.Instance.AddCoins(Database.Instance.coinCount);

```

```

        SaveSystem.Instance.Saves.availableLabyrinthSizes[SaveSystem.Instance.Saves.chooosedLabyrinthSize].count++;
    }

    public void InstanceWinStar(int starCount)
    {
        StartCoroutine(InstanceWinStarEnumerator(starCount));
    }

    private IEnumerator InstanceWinStarEnumerator(int starCount)
    {
        Database.Instance.isWin = true;
        yield return new WaitForSeconds(0.2f);

        switch (starCount)
        {
            case 0:
                Database.Instance.winBadStars[0].SetActive(true);
                yield return new WaitForSeconds(0.3f);
                Database.Instance.winBadStars[1].SetActive(true);
                yield return new WaitForSeconds(0.3f);
                Database.Instance.winBadStars[2].SetActive(true);
                break;
            case 1:
                Database.Instance.winGoodStars[0].SetActive(true);
                yield return new WaitForSeconds(0.3f);
                Database.Instance.winBadStars[1].SetActive(true);
                yield return new WaitForSeconds(0.3f);
                Database.Instance.winBadStars[2].SetActive(true);
                break;
            case 2:
                Database.Instance.winGoodStars[0].SetActive(true);

```

```

        yield return new
WaitForSeconds(0.3f);

Database.Instance.winGoodStars[1].SetActive(true);
        yield return new
WaitForSeconds(0.3f);

Database.Instance.winBadStars[2].SetActive(true);
        break;

        case 3:

Database.Instance.winGoodStars[0].SetActive(true);
        yield return new
WaitForSeconds(0.3f);

Database.Instance.winGoodStars[1].SetActive(true);
        yield return new
WaitForSeconds(0.3f);

Database.Instance.winGoodStars[2].SetActive(true);
        yield return new
WaitForSeconds(0.3f);

        break;

        default:

Database.Instance.winGoodStars[0].SetActive(true);
        yield return new
WaitForSeconds(0.3f);

Database.Instance.winGoodStars[1].SetActive(true);
        yield return new
WaitForSeconds(0.3f);

Database.Instance.winGoodStars[2].SetActive(true);
        break;
    }

    yield return null;
}

#endregion
}

```

## ItemsCreator.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ItemsCreator : MonoBehaviour {

    public SaveVariablesType type;
    public SliderPosition slider;
    public GameObject colorPanel;
    public GameObject pickerPrefab;
    public ColorPicker colorPicker;
    public ColorPickerTester colorPickerTester;
    public GameObject acceptPanel;
    public GameObject noMoneyPanel;

    private Database dataBase;
    private List<int> availableParts;
    private List<PartsDescriptionData> allParts;
    private int partNumber;
    private int clothesTypeNumber;
    private GameObject item;
    private List<ClothesPart> clothesList;
    private int firstSelectedItem;

    private Color choosedColor;//если будет выбор
    цвета то вот готовая переменная

    void Start ()
    {
        dataBase =
GameObject.FindWithTag("Database").GetComponent<Database>();
        clothesList = new List<ClothesPart>();
        ChoosePartList();
        CreateItems();
        slider.StartSlider();
    }

    private void ChoosePartList()
    {
        availableParts = new List<int>();
        allParts = new List<PartsDescriptionData>();

        if (type == SaveVariablesType.Body)
        {
            clothesTypeNumber = 0;
        }
        else if (type == SaveVariablesType.Hat)
        {

```

```

        clothesTypeNumber = 1;
    }
    else if (type == SaveVariablesType.Top)
    {
        clothesTypeNumber = 2;
    }
    else if (type == SaveVariablesType.Down)
    {
        clothesTypeNumber = 3;
    }
    else if (type == SaveVariablesType.Shoose)
    {
        clothesTypeNumber = 4;
    }
    else if (type == SaveVariablesType.Glasses)
    {
        clothesTypeNumber = 5;
    }
}

availableParts =
dataBase.saveSystem.Saves.availableCharacterParts[clothesTypeNumber].availableParts;
allParts =
dataBase.partsData.characterParts[clothesTypeNumber].parts;
availableParts.Sort();
}

private void CreateItems()
{
    ClothesPart clothesPart;
    PartsDescriptionData partsDescription;

    for (int i = 0; i < allParts.Count; i++)
    {
        int number = i;
        partsDescription =
dataBase.partsData.characterParts[clothesTypeNumber].parts[i];
        item = Instantiate(dataBase.itemPrefab,
transform) as GameObject;
        clothesPart =
item.GetComponent<ClothesPart>();

        clothesPart.itemCreator = this;

        clothesPart.color =
dataBase.saveSystem.Saves.availableCharacterParts[clothesTypeNumber].partsColor[number];

        clothesPart.clothesTypeNumber =
clothesTypeNumber;
        clothesPart.partNumber = i;
        clothesPart.clothesTypeNumber =
clothesTypeNumber;
        clothesPart.mainImage.sprite =
partsDescription.sprite;

        clothesPart.mainButton.onClick.AddListener(delegate { slider.StartSetPositionCoroutine(number); });

        clothesPart.selectButton.onClick.AddListener(delegate { slider.StartSetPositionCoroutine(number); });

        clothesPart.buyButton.onClick.AddListener(delegate { slider.StartSetPositionCoroutine(number); });

        clothesPart.buyButton.onClick.AddListener(delegate { clothesPart.BuyItem(); });

        clothesPart.mainButton.onClick.AddListener(delegate {
dataBase.currentPlayer.WearCloth(clothesTypeNumber,number); });

//clothesPart.colorBtn.onClick.AddListener(delegate {
clothesPart.InitializingColorPicker(clothesTypeNumber); });

        if (Utility.IsInList(availableParts, i))
        {
            if (i ==
dataBase.saveSystem.Saves.availableCharacterParts[clothesTypeNumber].choosedParts)
            {
                firstSelectedItem = i;

                clothesPart.selectObject.gameObject.SetActive(true);

                clothesPart.selectButton.gameObject.SetActive(false);

                clothesPart.buyButton.gameObject.SetActive(false);

                if (clothesTypeNumber == 0)
dataBase.tempBody = i;
                else if (clothesTypeNumber ==
1)dataBase.tempHat = i;
                else if (clothesTypeNumber ==
2)dataBase.tempTop = i;
                else if (clothesTypeNumber ==
3)dataBase.tempDown = i;
                else if (clothesTypeNumber ==
4)dataBase.tempShoose = i;
                else if (clothesTypeNumber ==
5)dataBase.tempGlasses = i;

            }
            else
            {

```

```

clothesPart.selectButton.gameObject.SetActive(true);
clothesPart.selectObject.gameObject.SetActive(false);
clothesPart.buyButton.gameObject.SetActive(false);
    }
    else
    {
clothesPart.buyButton.gameObject.SetActive(true);
clothesPart.selectObject.gameObject.SetActive(false);
clothesPart.selectButton.gameObject.SetActive(false);

        if (partsDescription.currencyType ==
CurrencyType.Coin)
        {
            clothesPart.currencyImage.sprite =
dataBase.coinImage;
        }
        else
        {
            clothesPart.currencyImage.sprite =
dataBase.diamondImage;
        }

        clothesPart.priceText.text =
partsDescription.price.ToString();

```

```

        clothesPart.price = partsDescription.price;
        clothesPart.currencyType =
partsDescription.currencyType;
    }

    clothesList.Add(clothesPart);
}

public void SelectNewItem(int partNum)
{
    partNumber = partNum;

    for (int i = 0; i < clothesList.Count; i++)
    {
        if (i == partNum) continue;

        if (Utility.isInList(availableParts, i))
        {
clothesList[i].selectButton.gameObject.SetActive(true);
clothesList[i].selectObject.gameObject.SetActive(false);
        }
    }

    public void SelectOldItem()
    {
        clothesList[firstSelectedItem].SelectItem();

        slider.StartSetPositionCoroutine(firstSelectedItem);
    }
}

```

## LabyrinthGenerator.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LabyrinthGeneration : MonoBehaviour
{

    private GameObject cellsMain;
    private Point startPoint, endPoint;

    void Start()
    {

```

```

        cellsMain = Database.Instance.cellsMain;

        SetSomeVariables();
        StartGeneration();
    }

    public void StartGeneration()
    {
        StartCoroutine(Generation());
    }

    private IEnumerator Generation()

```

```

{
    yield return new WaitForEndOfFrame();

Grid.Instance.MapGeneration(Database.Instance.m
apSize, Database.Instance.sizeOfCell);

    yield return new WaitForEndOfFrame();
    MazeSpawner.Instance.GenerateMaze();
    yield return new WaitForEndOfFrame();

MazeSpawner.Instance.ConverThisToMapCell();

    yield return new WaitForEndOfFrame();
    RoomGenerated();
    yield return new WaitForEndOfFrame();
    SetMainPrefabs();

Grid.Instance.FindMinWay();

    yield return new WaitForEndOfFrame();
    CoinGeneration();

    yield return new WaitForEndOfFrame();
    ChestGeneration();

    yield return new WaitForEndOfFrame();
    Database.Instance.endLabGen = true;

    Music.Instance.SelectMusic();

    GameObject player =
Instantiate(Database.Instance.playerPrefab) as
GameObject;

    player.gameObject.transform.position = new
Vector3(Grid.Instance.Map[startPoint.x,
startPoint.y].xCell,
Database.Instance.playerPrefab.transform.position.
y, Grid.Instance.Map[startPoint.x,
startPoint.y].yCell);

    yield return new WaitForEndOfFrame();
    GameHelper.Instance.StartTimer();

    Debug.Log("End Gen");
    yield return null;
}

private void SetMainPrefabs()
{
    startPoint.x = 0;
    startPoint.y = (Database.Instance.mapSize - 1)
/ 2;

    endPoint.x = Database.Instance.mapSize - 1;
    endPoint.y = (Database.Instance.mapSize - 1) /
2;

    GameObject cell =
Instantiate(Database.Instance.cellStartPrefab,
cellsMain.transform) as GameObject;

    cell.gameObject.transform.position = new
Vector3(Grid.Instance.Map[startPoint.x,
startPoint.y].xCell, 0,
Grid.Instance.Map[startPoint.x, startPoint.y].yCell);

    cell =
Instantiate(Database.Instance.cellEndPrefab,
cellsMain.transform) as GameObject;

    cell.gameObject.transform.position = new
Vector3(Grid.Instance.Map[endPoint.x,
endPoint.y].xCell, 0,
Grid.Instance.Map[endPoint.x, endPoint.y].yCell);
}

private void SetSomeVariables()
{
    Database.Instance.endLabGen = false;
    SaveInforms si = SaveSystem.Instance.Saves;

    if (si != null) Utility.SetSeed(si.seed);
    else Utility.SetSeed(Utility.Next(0,
100000000));

    Database.Instance.style =
si.choosedLabyrinthStyle + 1;

    if (si != null) Database.Instance.mapSize = 5 +

```



```

(si.choosedLabyrinthSize * 4);
    Database.Instance.isWin = false;
    GameHelper.Instance.isPause = false;
    Database.Instance.allRooms = new
List<RoomDescription>();
    Database.Instance.coinCount = 0;
    Database.Instance.allCoins = new
MeshRenderer[Database.Instance.mapSize,
Database.Instance.mapSize];
    Database.Instance.visibleCoins = new
List<Point>();
}

private GameObject
InstantiatePrefab(PrefabInform prefabInform, int
turn, Point3D instPoint, Transform mainTransform)
{
    GameObject labElement =
Instantiate(prefabInform.prefab, mainTransform) as
GameObject;
    labElement = Utility.prefabTurn(labElement,
prefabInform, turn);

    if (turn == 1)
    {
        float temp =
prefabInform.additionalPosition.z;
        prefabInform.additionalPosition.z =
prefabInform.additionalPosition.x;
        prefabInform.additionalPosition.x = temp;

        prefabInform.additionalPosition.z *= -1;
    }
    else if (turn == 2)
    {
        prefabInform.additionalPosition.x *= -1;
        prefabInform.additionalPosition.z *= -1;
    }
    else if (turn == 3)
    {
        float temp =
prefabInform.additionalPosition.z;
        prefabInform.additionalPosition.z =
prefabInform.additionalPosition.x;
        prefabInform.additionalPosition.x = temp;

        prefabInform.additionalPosition.x *= -1;
    }

    return labElement;
}

public void GenerateBorders()
{
    int wallNumber = 2;
    for (int x = 0; x <
Grid.Instance.Map.GetLength(0); x++)
    {
        for (int y = 0; y <
Grid.Instance.Map.GetLength(1); y++)
        {
            for (int t = 0; t < 4; t++)
            {
                if (Grid.Instance.Map[x, y].sides[t])
                {
                    InstantiatePrefab(Database.Instance.labElements[w
allNumber], t, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),
Database.Instance.roomsMain.transform);
                }
            }
        }
    }
}

```

```

    }

    if (x == 0)
    {
        InstantiatePrefab(Database.Instance.labElements[w
allNumber], 2, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),

        Database.Instance.roomsMain.transform);
    }

    if (y ==
Grid.Instance.Map.GetLength(1) - 1)
    {
        InstantiatePrefab(Database.Instance.labElements[w
allNumber], 3, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),

        Database.Instance.roomsMain.transform);
    }
}

}

}

}

}

public void RoomGenerated()
{
    GameObject room;
    GameObject collider;

    for (int x = 0; x <
Grid.Instance.Map.GetLength(0); x++)
    {
        for (int y = 0; y <
Grid.Instance.Map.GetLength(1); y++)
        {
            List<int> freeSides = GetFreeSides(x, y);

            if (freeSides.Count == 1)//U - element
            {
                room =
InstantiatePrefab(Database.Instance.roomsStyle[Da
tabase.Instance.style].rooms[0], freeSides[0], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

                Database.Instance.roomsMain.transform);

                collider =
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[0], freeSides[0], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

                Database.Instance.roomsMain.transform);//collider

                collider.transform.parent =
room.transform;

                room.name =
Database.Instance.allRooms.Count.ToString();

                room.GetComponent<RoomDescription>().turn =
freeSides[0];

                Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
            }
            else if (freeSides.Count == 2)
            {
                if (AreTurnsNeighbors(freeSides[0],
freeSides[1]))//L element
                {
                    if (freeSides[0] == 0 &&
freeSides[1] == 3)
                    {
                        room =
InstantiatePrefab(Database.Instance.roomsStyle[Da
tabase.Instance.style].rooms[1], freeSides[1], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

                        Database.Instance.roomsMain.transform);//turn 3

                        collider =
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[1], freeSides[1], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

                        Database.Instance.roomsMain.transform);//collider

                        collider.transform.parent =

```

```

room.transform;

        room.name
Database.Instance.allRooms.Count.ToString();

room.GetComponent<RoomDescription>().turn =
freeSides[1];

Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
    }
    else
    {
        room
InstantiatePrefab(Database.Instance.roomsStyle[Da
tabase.Instance.style].rooms[1], freeSides[0], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);

        collider
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[1], freeSides[0], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);//collider

        collider.transform.parent
room.transform;

        room.name
Database.Instance.allRooms.Count.ToString();

room.GetComponent<RoomDescription>().turn =
freeSides[0];

Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
    }
    else// | - element
    {
        room
InstantiatePrefab(Database.Instance.roomsStyle[Da
tabase.Instance.style].rooms[2], freeSides[0], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);

        collider
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[2], freeSides[0], new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);//collider

        collider.transform.parent
room.transform;

        room.name
Database.Instance.allRooms.Count.ToString();

room.GetComponent<RoomDescription>().turn =
freeSides[0];

Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
    }
    else if (freeSides.Count == 3)//T element
    {
        if(freeSides[0] == 0 && freeSides[1]
== 1 && freeSides[2] == 2)
        {
            room
InstantiatePrefab(Database.Instance.roomsStyle[Da
tabase.Instance.style].rooms[3], 0, new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);

            collider
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[3], 0, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);//collider

            collider.transform.parent
room.transform;

            room.name
Database.Instance.allRooms.Count.ToString();

room.GetComponent<RoomDescription>().turn =
0;

Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
        }
    }
}

```

```

        else if (freeSides[0] == 1 &&
freeSides[1] == 2 && freeSides[2] == 3)
        {
            room =
InstantiatePrefab(Database.Instance.roomsStyle[Da
atabase.Instance.style].rooms[3], 1, new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);

            collider =
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[3], 1, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);//collider

            collider.transform.parent =
room.transform;

            room.name =
Database.Instance.allRooms.Count.ToString();

room.GetComponent<RoomDescription>().turn =
1;

Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
        }

        else if (freeSides[0] == 0 &&
freeSides[1] == 2 && freeSides[2] == 3)
        {
            room =
InstantiatePrefab(Database.Instance.roomsStyle[Da
atabase.Instance.style].rooms[3], 2, new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);

            collider =
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[3], 2, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);//collider

            collider.transform.parent =
room.transform;

            room.name =
Database.Instance.allRooms.Count.ToString();

```

```

room.GetComponent<RoomDescription>().turn =
2;

Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
        }

        else if (freeSides[0] == 0 &&
freeSides[1] == 1 && freeSides[2] == 3)
        {
            room =
InstantiatePrefab(Database.Instance.roomsStyle[Da
atabase.Instance.style].rooms[3], 3, new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);

            collider =
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[3], 3, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);//collider

            collider.transform.parent =
room.transform;

            room.name =
Database.Instance.allRooms.Count.ToString();

room.GetComponent<RoomDescription>().turn =
3;

Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
        }
    }

    else if (freeSides.Count == 4)// + element
    {
        room =
InstantiatePrefab(Database.Instance.roomsStyle[Da
atabase.Instance.style].rooms[4], 0, new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),

Database.Instance.roomsMain.transform);

        collider =
InstantiatePrefab(Database.Instance.roomsStyle[0].
rooms[4], 0, new Point3D(Grid.Instance.Map[x,
y].xCell, 0, Grid.Instance.Map[x, y].yCell),

```

```

Database.Instance.roomsMain.transform);//collider
    collider.transform.parent =
room.transform;
    room.name =
Database.Instance.allRooms.Count.ToString();
room.GetComponent<RoomDescription>().turn =
0;
Database.Instance.allRooms.Add(room.GetCompo
nent<RoomDescription>());
    }
    }
}

private List<int> GetFreeSides(int x, int y)
{
    List<int> freeSides = new List<int>();

    for (int i = 0; i < 4; i++)
    {
        if (!Grid.Instance.Map[x, y].sides[i])
freeSides.Add(i);
    }

    return freeSides;
}

private bool AreTurnsNeighbors(int turn1, int
turn2)
{
    if (Utility.PlusTurn(turn1) == turn2 ||
Utility.MinusTurn(turn1) == turn2) return true;
    else return false;
}

private void TowerGeneration()
{
    Database.Instance.allTowers = new
MeshRenderer[Database.Instance.mapSize + 1,
Database.Instance.mapSize + 1];
    for (int x = 0; x < Database.Instance.mapSize;
x++)
    {
        for (int y = 0; y <
Database.Instance.mapSize; y++)
        {
            if (x == Database.Instance.mapSize - 1
&& y == Database.Instance.mapSize - 1)
            {
                SaveTower(0, new Point(x, y), new
Point(x + 1, y + 1));
                SaveTower(1, new Point(x, y), new
Point(x + 1, y));
                SaveTower(2, new Point(x, y), new
Point(x, y));
                SaveTower(3, new Point(x, y), new
Point(x, y + 1));
            }
            else if (y == Database.Instance.mapSize -
1)
            {
                SaveTower(2, new Point(x, y), new
Point(x, y));
                SaveTower(3, new Point(x, y), new
Point(x, y + 1));
            }
            else if (x == Database.Instance.mapSize -
1)
            {
                SaveTower(1, new Point(x, y), new
Point(x + 1, y));
                SaveTower(2, new Point(x, y), new
Point(x, y));
            }
            else
            {
                SaveTower(2, new Point(x, y), new
Point(x, y));
            }
        }
    }
}

```

```

    }
}

private void SaveTower(int turn, Point
mapSpawnPoint, Point listPoint)
{
    GameObject tower;

    tower
InstantiatePrefab(Database.Instance.roomsStyle[Da
tabase.Instance.style].rooms[5], turn,

    new
Point3D(Grid.Instance.Map[mapSpawnPoint.x,
mapSpawnPoint.y].xCell, 0,
Grid.Instance.Map[mapSpawnPoint.x,
mapSpawnPoint.y].yCell),
Database.Instance.roomsMain.transform);

    Database.Instance.allTowers[listPoint.x,
listPoint.y] =
tower.GetComponent<MeshRenderer>();
}

private void CoinGeneration()
{
    GameObject coin;

    int count = 0;

    for (int x = 0; x < Database.Instance.mapSize;
x++)
    {
        for (int y = 0; y <
Database.Instance.mapSize; y++)
        {
            if (Grid.Instance.Map[x,
y].distanceFromStart % 3 == 0 &&
Grid.Instance.Map[x, y].distanceFromStart != 0
&& !Grid.Instance.Map[x, y].isFinish)
            {
                coin =
InstantiatePrefab(Database.Instance.items[2], 0,

                new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),
Database.Instance.roomsMain.transform);

                Database.Instance.allCoins[x, y] =

```

```

coin.GetComponent<MeshRenderer>();

                count++;
            }
        }
    }

private void ChestGeneration()
{
    int dist = 0;
    int x = 0, y = 0;

    for (int i = 0; i < Database.Instance.mapSize;
i++)
    {
        for (int j = 0; j <
Database.Instance.mapSize; j++)
        {
            if (Grid.Instance.Map[i,
j].distanceFromStart > dist &&
!Grid.Instance.Map[i, j].isFinish &&
!Grid.Instance.Map[i, j].isMainWay)
            {
                dist = Grid.Instance.Map[i,
j].distanceFromStart;

                x = i;
                y = j;
            }
        }

        for (int i = 0; i <
Database.Instance.allRooms.Count; i++)
        {
            if (Database.Instance.allRooms[i].position.x
== x && Database.Instance.allRooms[i].position.y
== y)
            {
                GameObject chest =
InstantiatePrefab(Database.Instance.items[3],
Database.Instance.allRooms[i].turn,

                new
Point3D(Grid.Instance.Map[x, y].xCell, 0,
Grid.Instance.Map[x, y].yCell),
Database.Instance.roomsMain.transform); } } }

```